EUROPEAN SOUTHERN OBSERVATORY
Organisation Européenne pour des Recherches Astronomiques dans l'Hémisphère Austral
Europpäische Organisation für astronomische Forschung in der südlichen Hemisphäre

VERY LARGE TELESCOPE

VLT Data Flow System

GASGANO User's Manual

Doc. No. VLT-PRO-ESO-19000-1932

X.x

Date 15 July 2001

Prepared: P. Amico
N. Kornweibel
M. Zamparelli

Approved: M. Peron

Released: P. Quinn

Change Record

Issue/ Revision	Date	Section/Paragraph affected since last revision	Reason/Remarks
1.0	September 27, 1999		Preliminary Draft
1.4	February 26, 2001	All	Draft PSO Commissioning MAR2001
1.4	February 26, 2001	All	Revised version
1.5	July, 2001	All	Public release

1 *WELCOME TO GASGANO*

Gasgano is a Data File Organiser developed and maintained by the European Southern Observatory (ESO) to help its user community to manage and handle in a systematic way the astronomical data observed and produced within almost all the observatory facilities (e.g. the tool will not help for instance with SEST data). The tool is designed according to ESO specifications for fits and tfits files. A basic familiarity of the reader with ESO DFS and format standards is assumed throughout this document. For more information on this subject, refer to section 1.4.

1.1 Purpose & Scope

This document is the user's manual for the *Gasgano* ESO Data Flow System File Organiser, in short *Gasgano.*

This manual describes *Gasgano* Version 1.5, the first public release of the tool. The tool was distributed to internal ESO users for the first time at the end of 1999.

This manual does not contain an installation guide. Installation instructions for home users are maintained online on the *Gasgano* web pages (http://www.eso.org/gasgano). Installation instruction for ESO observatories on site installation instructions can be obtained from the DMD System Engineering Group (seg@eso.org).

1.1.1 Location of the latest version of this manual

The latest version of this manual will be available from the official *Gasgano* web site. (http://www.eso.org/gasgano).

1.2 Intended Audience

This manual is intended for *Gasgano* users. Users can be either internal ESO users or astronomers of the ESO community. *Gasgano* shows different features depending on the software environment in which it runs. Features available to internal ESO users may be masked, and therefore invisible, for external users. Throughout the document these features are described in separate sections, which are marked as "ESO" in a framed text.

This is not a technical manual. For technical details refer to the *Gasgano* DFS Organiser Design Document (VLT-PRO-19000-1930) or contact the ESO System Engineering Group (seg@eso.org).

1.3 Audience pre-requisites

Gasgano is a mouse driven graphical tool. Familiarity with the use of a mouse or some other graphical pointing device is assumed.

Familiarity with VLT Instruments, with ESO Fits file headers, with the definitions of run ids (OBS.PROG.ID) observation blocks (OBS.ID), observing templates and HIERARCH ESO keywords and, for internal users, with the VLT archive and databases structure is a must.

1.4 Reference documents and web pages

[1] GEN-SPE-19400-794/1.1/25-Nov-97 - *Data Interface Control Document*

[2]VLT-SPE-ESO-19000-0842/0.92/14-Aug-97 - *VLT Data Flow System, Architectural Design Document*

[3] VLT-PRO-ESO-19000-1930/1.0/27-Sep-99 – *VLT Data Flow System, Gasgano DFS File Organiser Design Document*

[4] For more information on science operations and the VLT data flow, point at http://www.eso.org/outreach/info-events/ut1fl/whitebook/wb100.html

[5] For the example using the eclipse library of data reduction routines (chapter 6): http://www.eso.org/eclipse/, http://www.eso.org/eclipse/eug/eug/eug.html, http://www.eso.org/eclipse/isaac; refer also to the online help of eclipse (isaacp man recipe_name)

[6] More information on the Real Time Display software: http://archive.eso.org/skycat/docs/rtd/

1.5 Glossary

1.5.1 General

Where appropriate, these definitions were adopted from the Data Interface Control Document ([XX]). Minor wording changes have been made since *Gasgano* is used at several ESO facilities, not just the VLT.

Acquisition Template (AT): an Observation Block component. An Acquisition Template is used to specify how a target is acquired by the telescope. It may also specify any preliminary instrument configuration steps (e.g. set rotator to specific angle). It can contain parameters for interactive as well as automatic acquisitions. This template may define a different instrument/detector configuration from the templates within the Observation Description. Each science Observation Block contains at most one Acquisition Template.

Calibration Observation Blocks: Observation Blocks used to acquire calibration data.

Data Flow System (DFS): a system that can handle the flow of data and information within certain ESO facilities, in particular, the VLT. It includes subsystems for proposal handling, observation handling, science and engineering data archiving, science data pipeline processing and handling and data quality control [2]. The DFS is being developed by the *ESO Data Management and Operations Division* (DMD).

Data Organizer (DO): software, part of the DFS, responsible for the preparation of the actual pipeline reduction. It handles in a dedicated database, the so-called Observation Summary Table, all the information describing the data products needed for the pipeline (raw frames, physical data and derived data extracted from the archive). This includes a set of FITS keywords and a set of variables characterizing the category to which the data belong. Based on an event handling rule which defines the calibration strategy, it reacts upon events (e.g. delivery of a raw frame by an instrument through the archive, completion of a template or of an observation block) and applies a hierarchy of Reduction Rules for creating a reduction plan for one or more data products (creating a set of Reduction Blocks to be executed in a given order).

Data Products: data files delivered to PI/Co-Is as result of the execution of ESO observing programs or setup information submitted to ESO in relation to observing programs. Data products include observation frames, observatory calibrations, meteorological and seeing measurements relevant to primary observations, etc.

Exposure: a synonym for the acquisition of a single data frame, typically resulting in a single FITS file.

Header Keywords: header keywords can be standard FITS (8 character long, e.g. `TELESCOP, MJD-OBS`), ESO standard (e.g. `HIERARCH ESO OBS PROG ID`) or *Gasgano* Meta –Keywords (see sections 8.2 and 8.3). Within *Gasgano* `HIERARCH ESO` keywords are specified in "dot" notation without the `HIERACH ESO` part. E.g.: `OBS.PROG.ID`.

Master calibration product: a reduced frame used as a master for calibration of science and calibration raw frames.

Observing Run: an approved ESO Program consists of one or more Observing Runs, each of which specify an independent combination of telescope, instrument, and observing operations mode (i.e. Service Mode or Visitor Mode).

Observing Program ID: see Run ID.

Observation: a coordinated sequence of telescope, instrument, and detector actions that results in a scientific or technical dataset.

Observation Blocks (OB): the smallest observational unit within the Data Flow System. An Observation Block contains a sequence of high level operations, called templates, that need to be performed sequentially and without interruption in order to ensure the scientific usefulness of an observation. Observation Blocks may contain only one target acquisition. Observation Blocks may contain scheduling and collect all status information of operations as they pass through the Data Flow System from definition, execution and reduction. Observation Blocks used in both Visitor and Service Mode to acquire data. They are used to acquire scientific, calibration, and technical data.

Observation Block ID: the unique identifier of an Observation Block. It is stored in the keyword `HIERARCH ESO OBS ID`. Valid OB ID examples: 19368, 200083938

Pipeline: a set of software tools, part of the DFS, which implements data reduction. Raw frames are classified, grouped by types and sent to data reduction recipes or reduction algorithms. Recipes produce pipeline products..

Raw Frame: The data product containing the result of an observation. Different instrument modes produce in general different observation frames.

Real Time Display: The Real Time Display software, or RTD was designed to display astronomical images, either from image files or from some external source, such as a CCD camera or external process via shared memory.

Reduction Block (RB): it is the equivalent of the Observation Block at the end of the data flow system chain. It defines the basic blocks needed for the data reduction software to process data. A reduction block contains: the name of the instrument, which produced the frames, the reduction recipe which has to be applied, the name of the output files (the products), the list of raw frames to be used as input to the reduction recipe and all parameters and master calibration files to be used as additional inputs. The reduction block is created by the DO software as an ASCII file with extension rb. After processing by the RBS software the extension is renamed to .rX (executed), rZ (failed, no product produced), .rY (some error in execution). The reduction block allows the application to associate the raw files and the master calibrations to the products generated by a recipe. When a reduction block file is not present, *Gasgano* is still able to visualize a so-called “virtual reduction block”, which works exactly as a real one and serves the same association purpose. The virtual reduction block is created by means of keywords (REC1 keywords) saved in the header of all data products. The virtual reduction block is a feature introduced during observing period 67 (in some test cases). Therefore data sets distributed before April 2001 are not compatible with it.

Reduction Block Scheduler (RBS): software, part of the DFS, which operates on Reduction Blocks, obtained from the Data Organizer: it analyzes the RB, checks the presence of input files and fires the corresponding data reduction recipe.

(Pipeline) Reduced Frame: the result of data reduction applied to either raw or calibration frames. Reduced frames are also called **pipeline products**.

Run ID: Same as Observing Program ID. the ID of an Observing Run. The format is PPP.X-0000(X). Valid Run ID examples: 63.A-0352(A), 167.A-4563(B). The Run ID is stored in the `HIERARCH ESO OBS PROG ID` keyword.

Service Mode (SM): observing operations mode where the astronomer submits a detailed description of their observing program to ESO for later possible execution. Service Mode programs are executed primarily in order of their ESO Observing Programs Committee assigned priority but only when the astronomer specified observing conditions are achieved onsite.

Science Observation Blocks: Observation Blocks used to acquire scientific data.

Template: a high-level data acquisition operation. Templates provide the means to group commonly used procedures into well-defined and standardized units. They can be used to specify a combination of detector, instrument, and telescope configurations and actions. Templates have input parameters described by a template signature, and produce results that can serve as input to other templates. As an example, an Acquisition Template takes target coordinates and produces through an interactive procedure the precise positions used later, e.g. to place an object on a slit.

Visitor Mode (VM): observing operations mode where the astronomer is present at the telescope while their observing program is being executed**.**

VLT Control System (VCS): the software and hardware tools that are used to control directly VLT instruments, telescopes, and related hardware. It enables and performs the acquisition of scientific and technical data. Originally designed for the Very Large Telescope, the VCS has also been installed at the La Silla New Technology Telescope (NTT) and 3.6m telescope.

1.6 Abbreviation and Acronyms

AT	Acquisition Template
DFS	Data Flow System
ESO	European Southern Observatory
GUI	Graphical User Interface
OB	Observation Block
RB	Reduction Block
RTD	Real Time Display
SM	Service Mode
UT	(VLT) Unit Telescope
VM	Visitor Mode
VLT	Very Large Telescope
VRB	Virtual Reduction Block
VCS	VLT Control System

1.7 Conventions used in this document

Features available for *Gasgano* running on ESO computers are framed. The label ESO appears on the top left.

- Tips are highlighted by the candle bullet (🕯 - we chose a candle because our tips are not that bright and you may get burnt)

- Examples and default values are typed using `Courier fonts.`
- *Gasgano* graphical user interface related items are typed in **bold**.
- Keystrokes combinations are typed in ***bold italic***.

1.8 How to get help or report problems with Gasgano or this manual

In all of the following situations:

- Gasgano bug reports
- Suggestion for improvements to this manual
- Technical assistance

contact the **ESO User Support Group (**usg-help@eso.org**).**

2 *OVERVIEW*

Gasgano is a GUI software tool for managing and viewing data files produced and handled by VLT Control System (VCS) and the Data Flow System (DFS)

Specifically, *Gasgano* scans a specified set of directories and organises the data in a manner familiar to its user, showing key information about the files and grouping the data primarily according to its Run ID and Observation Block ID. Once the data is presented to the user, it is available for viewing, archiving and reporting. Single files or groups of files can be selected by means of a single mouse click and sent as input to external scripts or software. Additional sort and search functions allow the user to refine his/her query or locate specific information.

ESO	In addition to this, *Gasgano* can be used as a database browser to easily locate and view data sets corresponding to a given observing program, or taken with a given instrument or a given telescope. This functionality is only available with ESO *observations* database.

2.1 What Gasgano does

The Gasgano tool:

1. provides a means to display sets of DFS data files grouped in a “virtual directory structure”, which reflects the program and observation grouping of the data files. The grouping of files allows easy identification of all data files associated with a `OBS PROG ID` and an `OBS ID`. An additional, instrument-specific level of keyword-based grouping can be added. *Gasgano* recognises fits files and fits tables and ASCII files by means of their file extensions (e.g. .fits and .tfits);
2. provide a means to classify files according to a user defined sets of rules. The classification of a file is stored in the *Gasgano* Meta-keyword `CLASSIFICATION;`
3. provides a clear mapping between raw fits files, pipeline products and master calibration files;
4. provides a mean to select files for,e.g., display of header keywords and image data
5. provides a mean to move/copy files to other directories;
6. provides a mean to generate reports for a set of selected files. These reports may be configured by definitin of list of header keywords to be printed in a tabular form.
7. provides a mean to select sets of files for archiving to standard applications: e.g. tar.
8. provides a mean to send selected list of files (by name) to external applications;
9. provides a mean to filter the file sets based on keyword expressions. File filters can be edited and saved, and loaded from disk;
10. provides a mean to classify files based on keyword expressions;
11. provides keyword expression functionality to define filter and classification rules . These expression include AND, OR, =, >, <, >=, <=, != and “*” operators;

12. provides a mean to interactively refresh the display of a file set, reflecting the addition of new files, modification of exiting files or removal of files from the file system;
13. provides a mean to search keywords within a displayed header;
14. provides a mean to search for files based on the entries displayed;
15. provides a mean to send one or more FITS files to image display software (e.g.: RTD).

16. ESO – provides a mean to connect to the ESO VLT Observations database and view its content organised by Run ID and OBS ID. It' is possible to restrict the view to specific observing periods (e.g. observing period 67, all Run IDS of the kind 67./167./267.X-0000(X)), to filter the views, to search for entries and to download files from the archive, provided the user has an account with the proper permissions set. If unsure of your account status and permissions, contact the VLT archive (catalog@eso.org)
17. ESO - provides a mean to complement the filesystem view with information on the OBS completion status from the obreb2 database, provided the user has an account with read permission to the database.

2.2 *Gasgano* at a glance

Figure 1 shows the main screen of the tool. It consists of three panels: the N**avigation Panel** on the top, the **File Association Panel** in the middle and the **File Detail Panel** on the bottom. Sliding bars separates the panels and allows panel resizing.

On the left-hand end of each panel bar there are two control buttons, of triangular shape, which allow expanding and collapsing of a panel, both upward and downward.

ESO The **Navigation Panel** consists of three different registers. In addition to the **Filesystem** view, there is a **Database** view and a **Results** view**.** The former allows connection to the ESO observations database and shows the files organised by OBS.PROG.ID and OBS.ID. The **Result** view, allow the user to browse an **info** file, as created by the data packer software, and shows sets of raw science files and their associated calibrations.

1. The **Navigation Panel** contains:

   - the **Menu bar** (**File, Selected Files, Tools** and **Help**)
   - the **Toolbar**. The T**oolbar**, from left to right, shows:
     - the **External Viewer button**, to start a viewing application (e.g. RTD) and display the selected file,
     - the **Frame Report button**, which creates custom reports,
     - the **script board button**, which launches the script board window, to run selected executables.

     The three buttons operate on selected files.
     - the **Grouping Menu button** and the **Expand/Collapse button,** which change the "virtual view" in the **Navigation Panel.**

- The **Find Entry** field accepts typed entries and the associated **find** button search for the entry in the **Navigation Panel**.

- **File counter:** on the right-hand end of the toolbar a number indicates the total number of files selected at any given moment.

ESO	Right of the number of selected files, the DO and RBS icons are shown. These icons are displayed in blue, whenever *Gasgano* detects that the Data Organiser and/or the Reduction Block Scheduler are currently running on the local workstation. They are displayed in white otherwise.

Files loaded by the tool are shown in the **Filesystem** view of the **Navigation Panel**, sorted by Run ID and OB ID, and organised in a directory tree, which is called a "virtual view". A selected list of keywords, configurable by the user, is listed. The keyword names are displayed at the top of each column.

2. The **File Association Panel** displays associations between set of raw frames, (pipeline) reduced frames and the master calibration frames used to produce them. Its top part is organised into four different registers:

   - **Raw Files**,
   - **Master Calibration Files**,
   - **Reduction Block**
   - **Product Files**.

   Clicking on an entry in any of these registers makes it the active selected file.

3. The **File Detail Panel** shows the header of a file selected in the **Navigation Panel.** This panel can be used to filter the header using a user-defined list of keywords and to search for strings in the header itself. Depending on the type of file selected, the **File Details Panel**, displays different information: the header for fits and tfits file, the content for ASCII files (including special ASCII files like the reduction blocks).

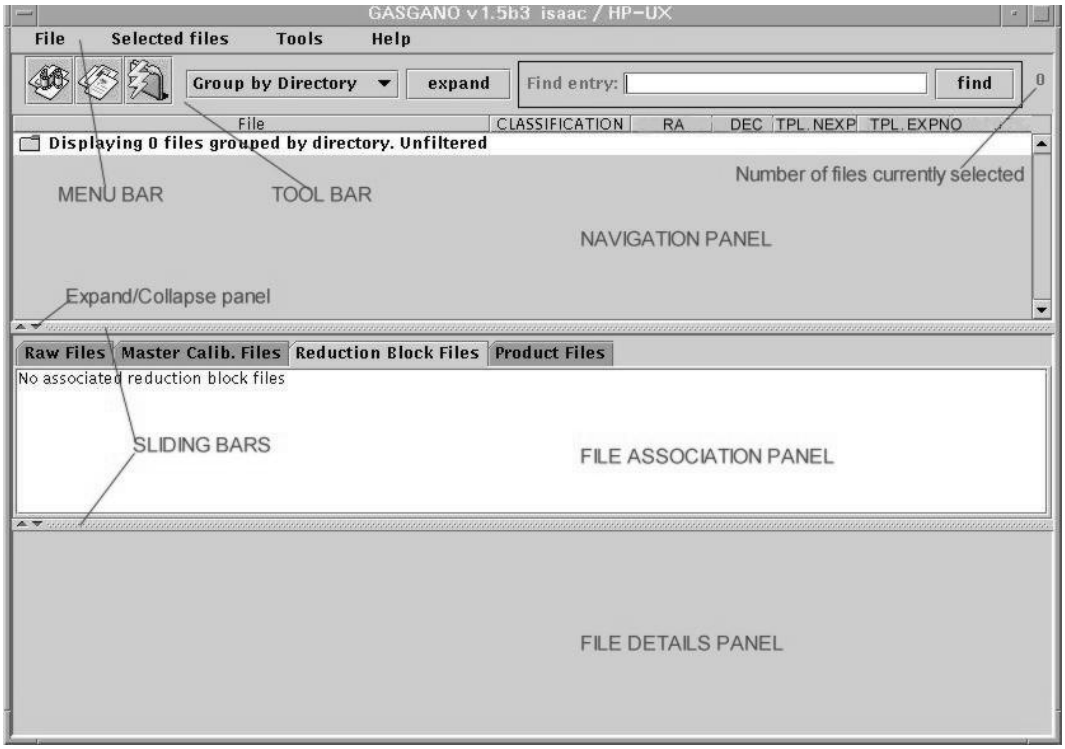

Figure 1: Gasgano's main panel. The three sections are, from top to bottom, the Navigation Panel, the Association Panel, the File Detail panel. Sections are separated by sliding bars, which can be moved up or down with "click and drag" or can be collapsed using the little triangular icon on their right side. The Menu bar contains File, Selected Files, Tools and Help

Gasgano's main use is that of a file browser. It can be used to browse through fits, tfits and ASCII files in a files system. It is optimised to work with ESO DFS standard files, i.e. files created with one of the VLT Unit telescopes and their instruments. Any file which is neither fits or tfits, that is, .fits and .tfits extensions, is ignored by the tool upon loading. The only exception to this rule are plain ASCII files, which are recognised, loaded and displayed provided that the tool is configured to recognise their extension. More on configuration can be found in chapter 4.

After a set of VLT standard files is loaded into the tool, (more on this in chapter 5) – the main panel resembles Figure 2. The **Navigation Panel** displays 98 files organised in folders, according to their Run ID (e.g. `67.D-0351(A)`), their `OBS ID` (e.g. `105201`) and optionally according to a user selected keyword (e.g. `TPL.START`, which groups all files observed within the same template execution). The Run ID folder line usually contains also the instrument name, taken from the `INSTRUME` keyword of the first file listed in the folder. This view is also called a "tree", because files are organised in a tree-like structure, with the `OBS.PROG.ID, OBS.ID` and `TPL.START` (in this example. Another keyword may be selected) as branches. The view can be modified in many ways (more on this in section 5.2). Each folder is displayed with a node icon, a folder icon and a line of text describing the folder (**folder line).** The latter may be, for instance, the combination of `the OBS.PROG.ID + INSTRUME` keywords, as seen in Figure 2. E.g. `60.A-9021(A) ISAAC.`

Clicking on any folder line or icon in the view "opens" it, or in other words, displays its content.

Raw files are displayed in blue, reduced in red, text files in grey. The icons for each file differ according to the file type. For each file the values for some selected keywords are displayed. In this example `CLASSIFICATION` (a *Gasgano* proprietary keyword – SPECTRA_COMBINED in the example), `DET.DIT (60.0000) DET.NDIT (1), INS.MODE (SWS1-…), INS.FILT1.ID (SK)` and `INS.GRAT.WLEN (2.1)` are displayed.

The file `IS_SSCM_010608A_SK_s03t_216_105201,fits` is selected, as can be seen by the light blue highlight band. On the upper right corner of the window the number `1` gives the total number of selected files. Since the selected file is a pipeline product (an ISAAC short wavelength science combined spectrum), the **Association Panel** shows its properties and associated files. In Figure 2 the **Master Calib. Files** register is displayed. Two frames, with their path in the file system, are listed. These frames are the master calibration frames used during the data reduction of the science spectrum. The IS_SSAM* file is the arc frame and the IS_ISSFM* file the spectroscopic flat. Both files are loaded in the data set. Clicking on one of these files in the **Association Panel**, will move the selection in the **Navigation Panel** to highlight it.

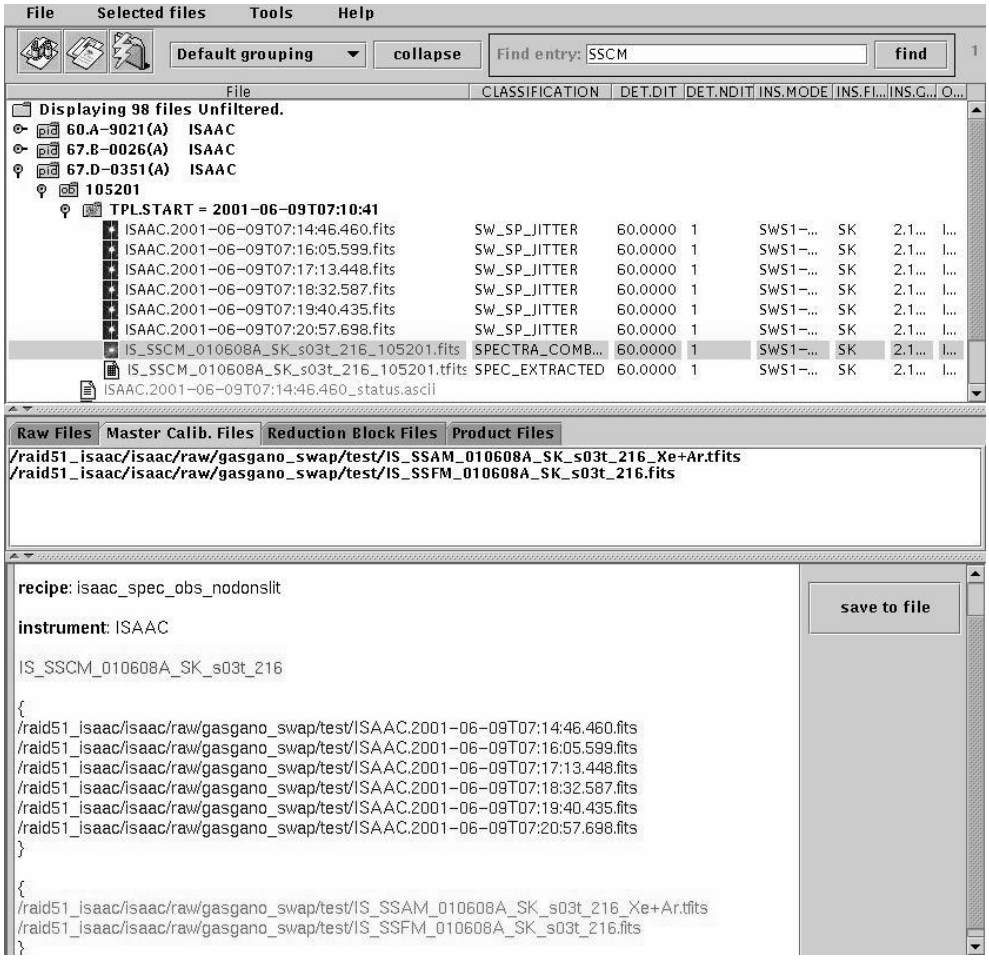

Figure 2: *Gasgano's* main panel after loading a typical VLT data set. In this specific example files observed with the VLT Antu telescope and ISAAC are shown. Refer to the text for a description of this figure.

The first row of the **Navigation Panel** displays a message indicating how many files are loaded, and whether or not they are grouped or filtered (more on grouping and filtering in Chapter 5 sections 5.2.1 and 5.4). The typical message is:

```
Displaying nnn files. Unfiltered.
```

The lower **File Detail Panel** displays the virtual reduction block associated with the science spectrum. As explained in the glossary, reduction blocks list all the "ingredients" used to create a reduced frame. The reduction block can be an ASCII file created by the Data Organiser, or can be virtual, that is created on the fly by *Gasgano* using header keywords information. In Figure 2 a virtual reduction block for the selected file is displayed in the **File Details** Panel. Its components are:

- the data reduction **recipe** used (`isaac_spec_obs_nodonslit`),
- the instrument (`ISAAC`), the basename of the data products: (`IS_SSCM_010608A_SK_s03t_216`),
- the set of raw files that were reduced (included in the firs set of curly braces),
- the optional set of master calibration frames (in the second set of curly braces), The latter set may also contain special parameters used for the data reduction.

The structure of the reduction block is always the same and by means of the save to file button on the right hand side of the **File Details** Panel it is possible to write the virtual reduction block to a file.

EEO - Once a virtual reduction block is saved to file, it can be loaded and handled by the tool. This for instance includes the possibility to request the RBS to execute it again, or the possibility to edit it, to modify it and then run it again. The re-execution of the RB cannot be triggered within *Gasgano*, unless, the RBS is running and the saved RB file is copied in the proper directory (usually $DFS_REDBLOCK).

3 *GASGANO INSTALLATION AND STARTUP*

3.1 System definition and requirements

Software requirements: supported platforms are Linux, HP-UX, SunOS with JDK-1.1.7 or 1.1.8. For viewing fits file, an image display software such as RTD or Saoimage is needed.

Hardware Requirements: *Gasgano* requires at least 100 Mb of RAM (depending upon data set size and usage) so it's advisable to run it on machines with not less than 512MB RAM. If visualised on a remote X11 terminal, make sure that at least 48 MBs of memory is available.

Processor speed becomes an issue only when accessing large numbers of compressed fits files.

ESO *Gasgano* is a Java based application shipped with ESO's DMD Data Flow System software. It is intended for use on DFS workstations and is targeted for these Unix platforms. This document assumes *Gasgano* is running in such an environment whenever it describes ESO features.

System Settings: since *Gasgano* is an X application, prior to launching, the `DISPLAY` environment variable must be correctly set. *Gasgano* will activate a'heartbeat' functionality for the processes *Data Organiser* and *Reduction Block Scheduler* if the environment variables `DFS_PRODUCT` and `DHS_DATA` are defined (that is a DFS pipeline environment is in place). This is used to monitor the Data Reduction Pipeline activities.

3.2 Downloading Gasgano

Gasgano can be downloaded from the dedicated Web pages, located at:

http://www.eso.org/gasgano.

Installation instructions are also posted on this web page.

Contact the User Support Group (usg-help@eso.org) if you need assistance.

ESO *Gasgano* is supported by the Software Engineering Group. Please contact SEG regarding installation, upgrades, etc… (seg@eso.org)

3.3 Running Gasgano

To start the application enter the following on the command line:

- `gasgano &`

If *Gasgano* starts correctly, the main screen will appear. If the `DISPLAY` setting is incorrect or `.Xauthority` is not correctly set, the application will not start, and messages similar to the following will be displayed on the screen:

```
Xlib: connection to “xt313:0.0” refused by server
Xlib: Client is not authorized to connect to Server
java.lang.InternalError: Can’t connect to X11 window server using
’xt313:0’ as the value of the DISPLAY variable.
at sun.awt.motif.MToolkit.<init>(MToolkit.java:103)
at java.awt.Toolkit.getDefaultToolkit(Toolkit.java:399)
at javax.swing.ImageIcon.<init>(ImageIcon.java:95)
at javax.swing.ImageIcon.<init>(ImageIcon.java:106)
at
at org.eso.gasgano.GasProp.loadProperties(GasProp.java:275)
at org.eso.gasgano.Gasgano.<init>(Gasgano.java:139)
at org.eso.gasgano.Gasgano.main(Gasgano.java:26)
```

3.4 Your local directory structure and configuration files

With the default installation, a directory `gasgano` is created under `${HOME}.` This directory contains all the executables and the accessory files. These files should never be modified directly by the user.

A *.gasganorc* configuration file will be created in ${HOME}.

The tool may use additional directories to store keyword lists for reports and filtering of headers (defaulted to `${HOME}/gasgano/keywords`), classification rules files, rules for file filtering (both defaulted to `${HOME}/gasgano/`), and user defined scripts (defaulted `${HOME}/gasgano/scripts`). The user may change the default values by means of the **Preference editor** (see section 4.1).

3.5 Exiting Gasgano

To exit *Gasgano*, select **File>Quit** on the Menu bar or type ***CTRL+Q***.

4 *CONFIGURING GASGANO – DO IT FIRST!*

Gasgano was designed to be flexible and configurable to the greatest possible extent in order to accommodate a wide range of uses. The tool comes with a standard configuration set, which can be tailored to specific user requirements by means of the **Preference Editor**.

Configuring the tool to suite your needs may take some time and effort, especially if you want to define your set of classification rules for the files. It is nevertheless **highly recommended** to do it before you plan to use the tool extensively. A good configuration setup is likely to save you a lot of time and, once it is done, you will be able to re-use it. Another advantage of configuring the tool first, is that you will become more familiar with its features.

4.1 Running the Preference Editor

To access the **Preference Editor**, click on **File>Preferences…**. The **Preference Editor** window will appear. It is organised in registers: **Data, Display and Printing, Menus and Instruments.**

ESO	In addition to the standard registers, an additional one, the **Databases** register**,** is available for database connectivity settings

Clicking on any of these registers activates a configurable panel. All panels have an **OK** and **Cancel** buttons on the lower part. Accepting the preferences via the **OK** button will have to twofold effect of saving them into the *.gasganorc* configuration file and enforcing them in the current session.

Though not required, the preference file can be edited using any ASCII editor. Refer to chapter 9 for the list of available entries.

4.1.1 Configuring the Data register

The Data register, shown in Figure 3, contains general file system settings: filenames of rules and filter files, recognised text files extensions and the location of the keyword list directory (the directory where all user-defined keyword lists are stored). The **Data register** is divided into three main sections:

1) In the upper part, the **Data Directories and Files** field specifies which directories or files are automatically loaded at start-up. Adding entries to this panel and accepting the preferences with the **OK** button automatically loads the corresponding data sets into memory, if not loaded yet. Selecting a directory file selects all files in the directory. To load a directory:

- click on the **Add File** button. A dialog box, which allows navigation through the file system, is opened.
- Select the desired directory or file and accept by clicking on the **Add File** button (a more detailed description of the **Add File** dialog box can be found in section 5.1).
- To remove a directory from the loaded list, select it with a single mouse click and then click on the **Remove Selected Files** button. When removing items from the list, multiple selections (***CTRL+left click*****)** are possible**.**

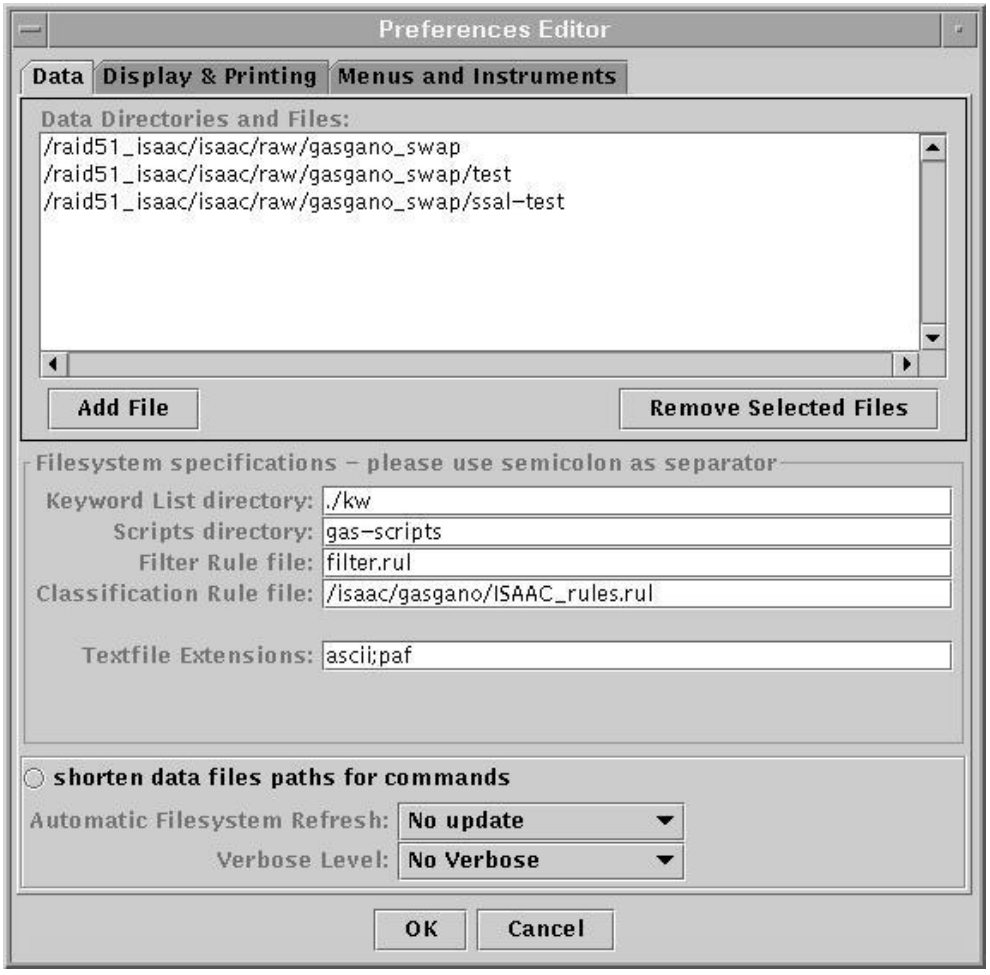

Figure 3: The **Data** register in the **Preferences…** Menu.

- In the middle section of the register, the **Filesystem** specifications allow the user to define:

  a. the directory where to store user-defined keyword lists, to be used, for example, to create printable reports or to filter the fits header displayed in the **File Detail** panel in the main screen. The default value is `${HOME}/gasgano/keywords;`

  b. the scripts directory, where to store user defined scripts or executables external command (see chapter 6). The default value is set to `${HOME}/gasgano/scripts`, under the account's `${HOME}` directory;

c. the name of the file which contains the rules to filter the files in the **Navigation Panel;** the default file is `filter.rul`, located into the account's `${HOME}/gasgano` directory;

d. the name of the file that contains the classification rules for the raw files. It is defaulted to `${HOME}/gasgano/rules.rul`.

e. the recognised text files extensions; it is defaulted to `ascii;paf; longlog,` but as many extensions as desired can be added. Entries should be separated with a semicolon.

Example: in the default case, the tool recognises `pippo.ascii` as a simple text file. When the file is loaded, if selected, its content is displayed in the **File Detail** panel. The file `pippo.txt` will not be recognised and therefore not loaded (ignored) by the tool.

When this list is modified, a manual refresh of the data set must be forced (Click on **File>Refresh** in the Menu bar). Moreover, if an extension is deleted from the list, the change takes effect the next time the tool is started or, in other words, the files with this extension are not automatically unloaded by the tool.

As already mentioned, *Gasgano* recognises file types by their file extension. For fits files these extensions are .fits (2-D frame) and .tfits (table).It is not possible to configure the file extensions for fits or tfits files. Therefore, the fits file `pippo.rtd` will not be recognised by the tool. The user should take care or renaming files to the default extensions.

- The bottom section deals with miscellaneous settings:

a. The **shorten data files paths for commands** button specifies whether the application should try to determine a common path for all data files supplied to an external command, and execute a `'cd'` to that path prior to executing the external command itself. This can be useful to overcome shell limitations in the maximum amount of characters permitted on the command line when calling an external application or scripts. If the user selects a set of files with more than one path, the file names are not shortened.

b. The **Automatic Filesystem Refresh** sets the frequency for a refresh procedure: the tool scans the loaded directories and verifies whether a file has been modified (and therefore its header needs to be read in again) or whether files have been added/removed from a directory in the current data set. The possible settings are: **No update** (default), **1,2,5,30 minutes**.

This option is useful when you are running data reduction procedures that create new files in the loaded directories or when you are moving files (from the command line) into/from loaded directories. This option should be turned off when you have loaded a large number of files, since the refresh freezes the application for an amount of time, which is roughly proportional to the number of files loaded.

c. The **Verbose Level** option is used for debugging purposes only and specifies which debug level should be monitored while the application is running. This option should always be set **to No Verbose**, which is the default value. For more information about the meaning of the other verbose levels, refer to chapter 10 of this manual.

- Whenever a verbose level is activated, diagnostic messages are written to stdout. When troubleshooting, your USG contact may ask you to activate one of the verbose levels and report these messages.

- It is recommended to start *Gasgano* with "`gasgano > my_log_file 2>&1 &`" if you want to reproduce an error. This command saves all messages from stdout and stderr in the file my_log_file, which can be used for troubleshooting

4.1.1 Configuring the Display & Printing register

The **Display & Printing** register, shown in Figure 4, contains settings for the **Navigation Panel** in the **Main Screen,** settings for the printer options and for configuring the file viewers.

The configurable items in this register are:

1) **Filesystem – Display Keywords** editable list: keywords in this list are displayed by the tool in the **Navigation Panel.** The display follows the list ordering. The following keywords may be listed:

   a. All 8 character standard fits and ESO keywords: Example: `RA,DEC, EXPTIME,ORIGFILE;`

   b. `HIERARCH ESO` specific keywords, without the `HIERARCH ESO` prefix. Example: `INS.FILT1.ID, TPL.START, INS.GRIS1.NAME;`

   c. *Gasgano* meta-keywords, which are defined by the tool only but are available for display purposes: `CLASSIFICATION, FILE_PATH` and `PIPE_PRODUCT`. More on these keywords and their meaning in section 8.3.

   - `CLASSIFICATION` is, among the meta-keywords, the most useful to display: it represents the classification of a raw science/calibration file done by the tool according to the classification rules, which are fully configurable by the user. More on classification rules and how to configure them in section 4.2.

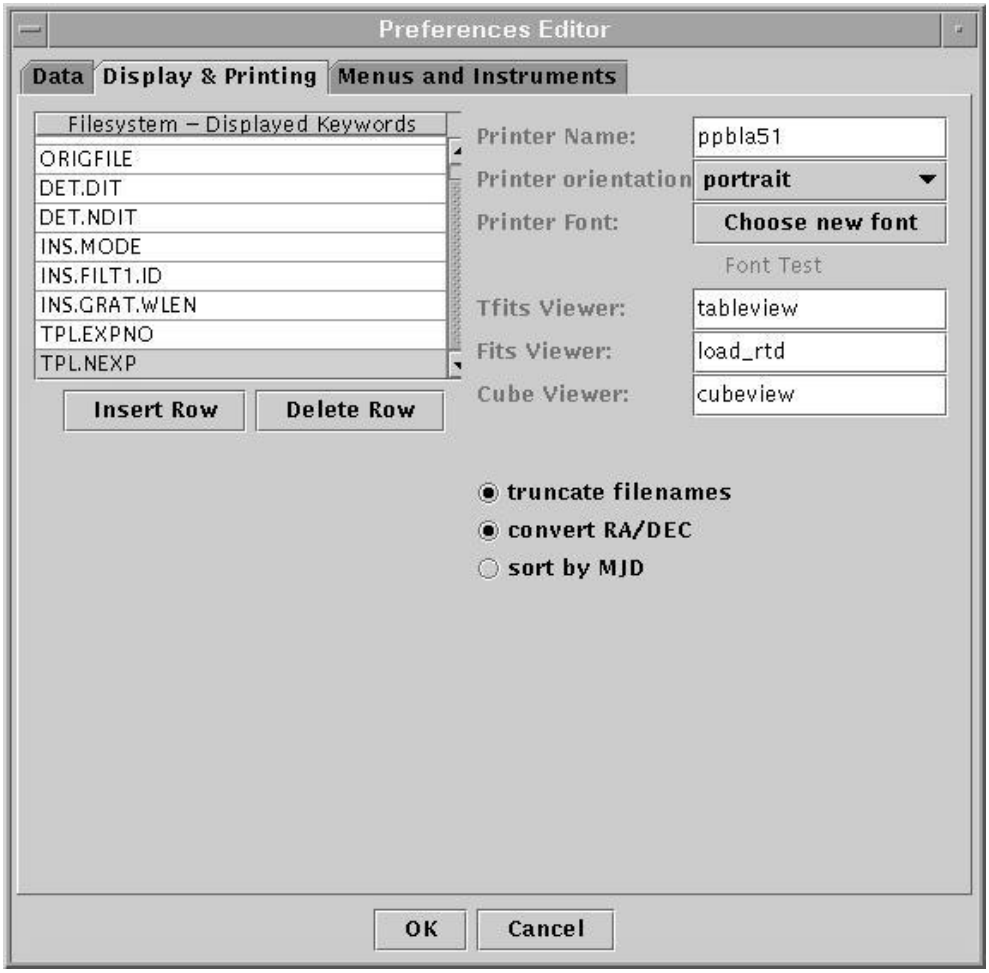

Figure 4: The **Display & Printing** register in the **Preferences...** Menu. The fields are filled in with possible entries. These entries are dependent on the system settings and user preferences.

To enter a keyword, type its name inside a field. The buttons **Insert Row** and **Delete Row**, allow the user to add more rows (a scroll bar appears on the left side, when there are more rows than those which fit in the field) or to delete selected row (no multiple selection is available). The keyword names are case sensitive (must be uppercase) and the tool will mark the field as `*INVALID*`, whenever this rule is not respected. There are no syntax or validity checks, therefore if a wrong keyword name is typed in, no error message is displayed, but there will be an empty column in the **Navigation Panel**. The order of the keywords in the list is the order of display of the column in the **Navigation Panel**.

To add a keyword above an existing one in the list, right-click on the **Insert Row** button.

2) The printer set-up fields, situated on the right half side of the register, include:

   a. **Printer Name**, with no default value. Specify in this field the "Unix" name of your printer; e.g. in Figure 4: `ppbla51`.

   b. **Printer orientation**, defaulted to `portrait`. It can be set to `Landscape`.

c. **Printer Font,** defaulted to `Monospaced, 8pts`. The chosen font can be seen below the field (`Font Test`). To change the font type, size and characteristics, click on the **Choose new font** button.

3) The viewers setup fields, located below the printer setup fields, allow specifying the UNIX command to launch file viewers. Three different viewers are possibly configurable:

a. **Tfits Viewer**, launched by the application whenever a single tfits file is selected;

b. **Fits Vie**wer**,** launched by the application whenever a single fits file is selected;

c. **Cube Viewer**, launched by the application whenever more than one fits file is selected;

It is responsibility of the user to define calls to external viewers. No viewers are shipped with *Gasgano* and the three fields are empty by default. It is possible to specify a viewer in two ways:

1) type the name of the application as you would from a terminal command line. The path to the application you plan to use should be in the `${PATH}`. Example: `rtd`. Note however that no "&" sign is allowed in the field and that in the case of the example, the application will freeze *Gasgano*. In order to be able to load one image after the other, clicking on the file name in the application panel, it's recommended using method 2)

2) You call an external shell script, which in turn calls another application.

4) The **truncate filenames** toggle on/off button allows the user to specify whether the filenames are displayed with their complete path or not in the Navigation Panel view. Default is on.

5) The **Convert RA/DEC** toggle button allows the user to specify whether the entries for right ascension (RA) and declination (DEC), if present in the **Displayed Keywords** list, should be displayed in sexagesimal or in decimal notation. Default value is **on,** sexagesimal.

6) The **Sort by MJD** toggle button, forces the application to display files sorted by the content of the keyword `MJD-OBS`. This option is useful if the files have been renamed following a user defined code, which would sort them in a different order than that of the UT time of observation, like is the case of the VLT Archive standard naming convention. Default value is **off**.

ESO 7) The **Database – Displayed Keywords** editable list: keywords in this list are displayed by the tool in the **Navigation Panel of the** Database view**.** What written under 1), is valid in this case, too.

4.1.2 Configuring the Menus and Instruments register

The **Menus and Instruments** register, shown **in** Figure 5**,** allows configuration of the virtual view in the **Navigation Panel** by adding an instrument-specific criteria to group files under common registers. The default virtual view, shows the loaded files organised in a "virtual" directory tree organised in run ID (`OBS.PROG.ID` keyword) and

OB ID (`OBS.ID` keyword) *intermediate "containers"* The user might wish to group files sharing some common properties, i.e. the same value for a specific fits keyword. The **Menus and Instruments** register contains an editable table, which contains two columns: the Instrument column lists all available instruments and the criteria column: both are editable. The **Instrument** column should contain the name of the instrument as specified by the `INSTRUME` keyword in the header. The user may want to specify for the **criteria a** header keyword, following the rules already explained in Subsection 2.2. Examples**:** `TPL.START,OCS.CON.QUAD`

For many instruments, and especially ISAAC, the use of `TPL.START` is particularly useful whenever an OB contains several templates. `TPL.START` is identical for all files belonging to the same template instance, and it is set to the UT starting time of its execution. For VIMOS, it may be convenient to group the files by the different detectors, making use of the keyword `OCS.CON.QUAD`. For FORS1 and FORS2, in case of observation with many different masks, grouping by means of `MASK.ID`, might prove useful. If the OBS.ID is the smallest "meaningful "container for your data set, and you want to group your files according to another keyword but on a higher level of grouping, it is preferable to make use of the file filtering options, described in section 5.4.

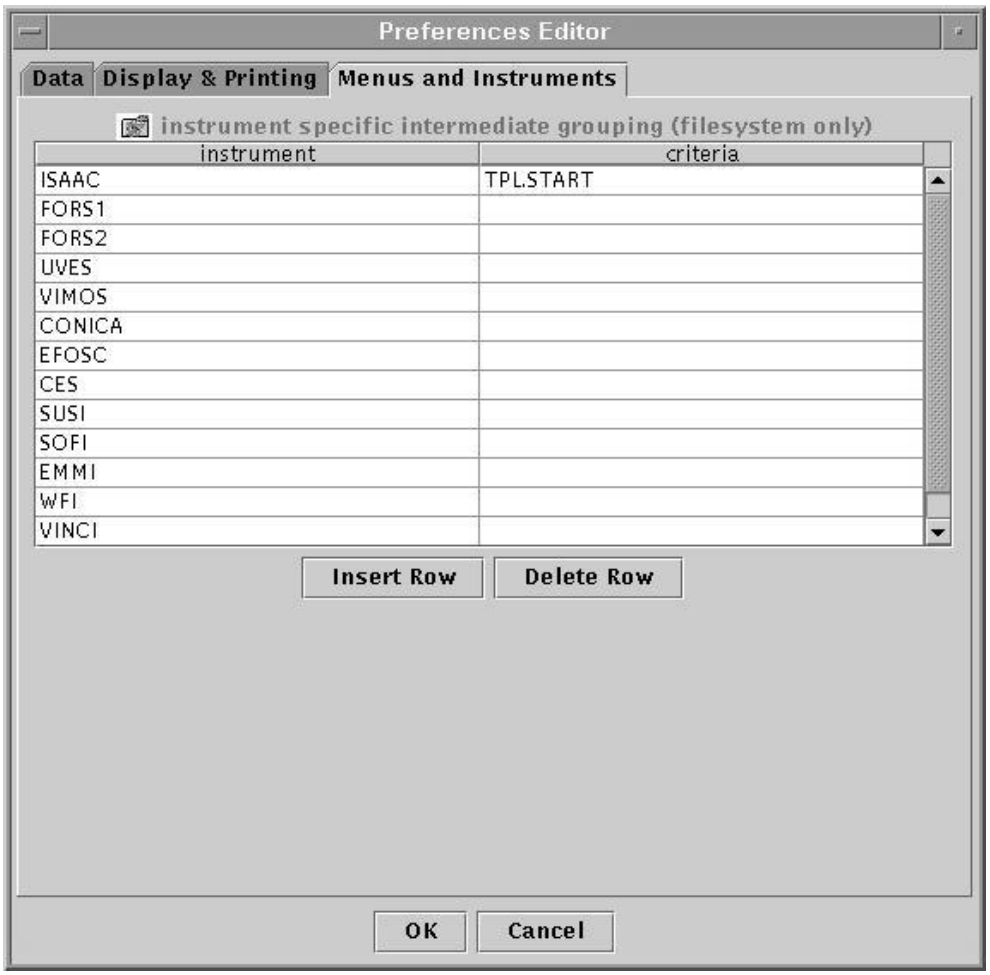

Figure 5: The **Menus and Instruments** register in the **Preferences…** Menu

4.1.3 ESO - Configuring the Databases register

You can skip this subsection and go to the next chapter if you are not an ESO internal user.

The **Databases** register, shown in Figure 6, allows specific settings related to the **Database view**. When Gasgano is started on an ESO workstation, the upper part of the Main Screen is divided into registers: in addition to the **Navigation Panel** of the **Filesystem View**, there is a **Navigation Panel** of the **Database View** and a **Results** register. Clicking on the register icon situated at the bottom left of the Navigation panel, a different register is selected. The Database view shows the content of the Observations Database in the ESO archive organised in a virtual view trees structure, identical to that offered for the file system (the files are organised by `OBS.PROG.ID` and OBS ID). More information on the database view in Chapter 7.

The use of the **database r**equires possessing an account with permission to access the obrep2 and observation databases.

The **Read observations from database toggle button activates the** Database **View in the** Main Panel. The fields below **allow** the user to specify the connectivity parameters for database access. In particular:

1) the server URL, in the format: `jdbc:sybase:Tds:<hostname>:<portnumber>;`

2) the server ID, simply for identification purposes. Example: `ESOECF;`

3) the username used to connect to the server;

Contact your Database Administrator for the correct values for the server you wish to connect to.

The **Display OB Status (from database)** toggle indicates whether the return status from the execution of an OB should be queried from the database and displayed in the application. The OB Status is then displayed in the **OBS.ID folder line**, after the `OBS.ID` number, in the **Navigation Panel.** If set to on, the fields below have to be filled with all the necessary information for database connectivity. Ask your database administrator or DMD personnel for assistance. The lower part of the panel contains a code-string association table for displaying purposes when the OB statuses from database are requested. Example: when the query to the database for a certain OB returns the value `X`, the **folder line** for that OB displays the message `executed`.

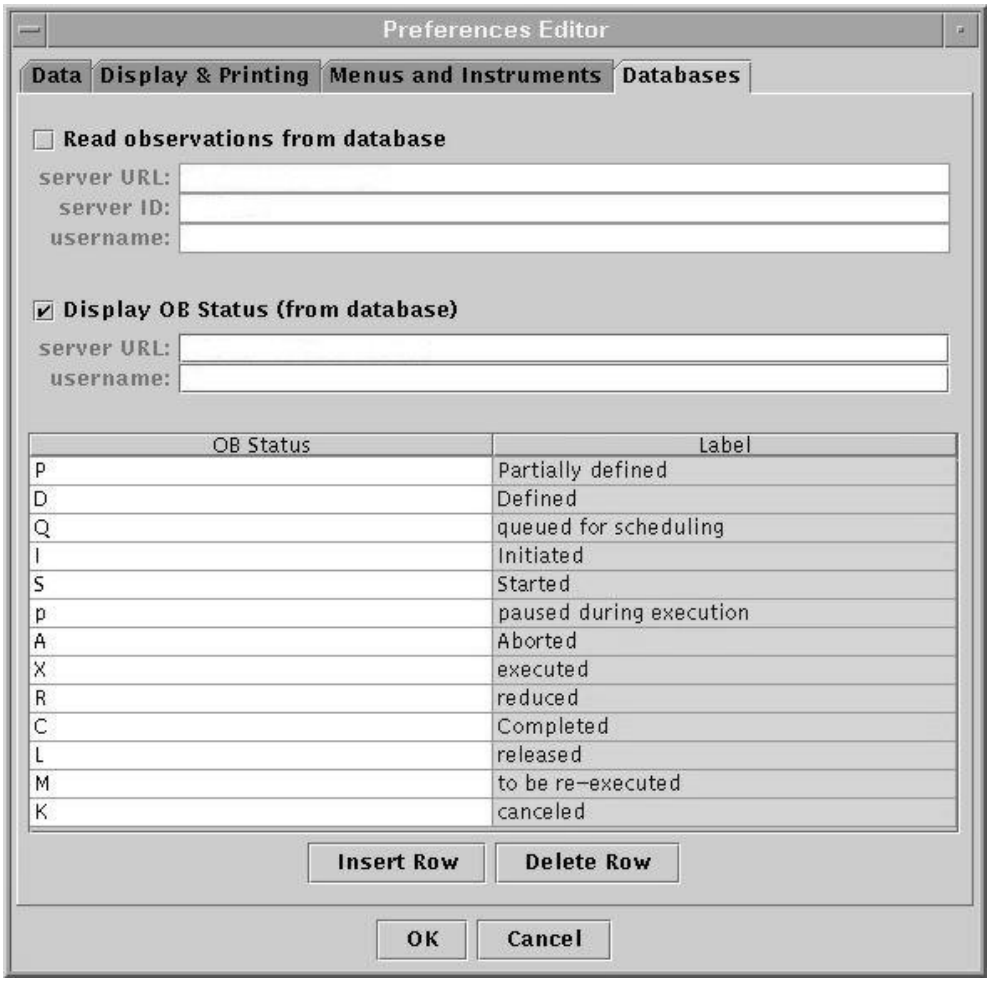

Figure 6: The **Databases** register in the **Preferences…** Menu of the *Gasgano* application for ESO users.

4.2 Editing the classification rules

Gasgano provides a mean to classify files based on keyword expressions. The concept of classification of files is used within the VLT DFS by the Data Organiser, which classifies files according to fits keyword related rules and then creates reduction block to be sent to the data reduction pipeline. Classification of files according to user specified rules might be useful for quick look assessment of a data package and for data reduction purposes.

Gasgano classifies files as they are loaded, comparing the value of specific keywords in their headers with the rules specified in the file (by default `${HOME}/gasgano/rules.rul`), which is located in the directory specified in the **Data** register of the **Preference Editor** (see subsection 4.1.1). The rules are applied in the order they are written in the classification rule file. If there is conflict between rules, identical or similar (intersecting) rules, the first in the file has the precedence. *Gasgano* is distributed to the users with a default classification rules file, which can be edited to meet the user's specific needs. This file contains basic classification rules for all the available VLT instruments and it is updated every time a new instrument becomes operational. In addition, instrument specific classification rules files are also distributed. Example: `ISAAC_rules.rul, UVES_rules.rul`, etc.

The use of the default files is encouraged for all first time users. More experienced users who wish to modify the existing rules and to create their own sets of rules are encouraged to read sections 4.2.2 and 4.2.3.

If the user foresees working with files produced by only a single instrument, it is advisable to configure *Gasgano* to use the appropriate instrument-specific classification rules file, instead of the generic one. It is also advisable to edit the file and delete or update the classification rules according to user's specific data reduction needs. The classification rules supplied with *Gasgano* are similar to those used by the ESO data reduction pipeline, therefore tailored to an automatic data reduction strategy for large amounts of data.

It is very important to spend some time configuring the classification rules. A good reason to do so has to do with the tool's performance. Assuming an average data packet of more than 500 files, the time needed by the application to load them into memory and to classify them is significantly higher the more classification rules are stored. This also applies whenever there is a need to reload the files or reclassify them (for instance if you change the columns displayed in the **Navigation Panel** in the **Main Screen)**. Selecting just the rules you require speeds up the classification of files.

The classification rules file is not in ASCII format and therefore can be edited from within the tool only, by means of the classification rules editor. To access the file click on the **Menu Bar Tools>Classification rules…** . The **Classification Rule Builder** window appears, as shown in Figure 7.

The window is divided into two sections, the list of rules in the upper part and the editor panel in the lower part. At the bottom, a set of buttons allows maintenance of the list of rules. A rule selected in the upper panel is displayed in the lower panel, where it can be edited. For each rule an instrument and a classification label must be specified. The instrument label tells the tool to apply a rule for files, whose keyword `INSTRUME` has the same value, the classification appears in the **Navigation Panel** if the `CLASSIFICATION` keyword is on the list of keywords to be **displayed (Preferences Panel, Display options** register).

The upper panel consists of a table with three columns: The **Instrument and classification** column display the content of the above mentioned labels. The rule column displays the classification criteria.

The lower panel allows display, creation and editing of classification rules. The rules are graphically rendered in a sort of **RPN (Reverse Polish Notation).** In short, RPN is a way of expressing arithmetic and/or logical expressions that avoids the use of brackets to define priorities for evaluation of operators. Here are some examples (not all of them are examples taken from the supplied classification rules files):

1) The simplest rule. The simplest rule classifies frames using a one keyword classification criteria:

```
ISAAC    TWFLAT     TPL.ID="ISAACSW_IMG_CAL_TWFLATS"
```

Where `ISAAC` is the instrument, `TWFLAT` is the classification label and the rule states that if in a fits file header the keyword `TPL.ID` (observing template

identity) is equal to ISAACSW_IMG_CAL_TWFLATS, then this file must be classified as TWFLAT (twilight flat). In the lower panel the rule will be displayed simply as:

```
TPL.ID="ISAACSW_IMG_CAL_TWFLATS"
```

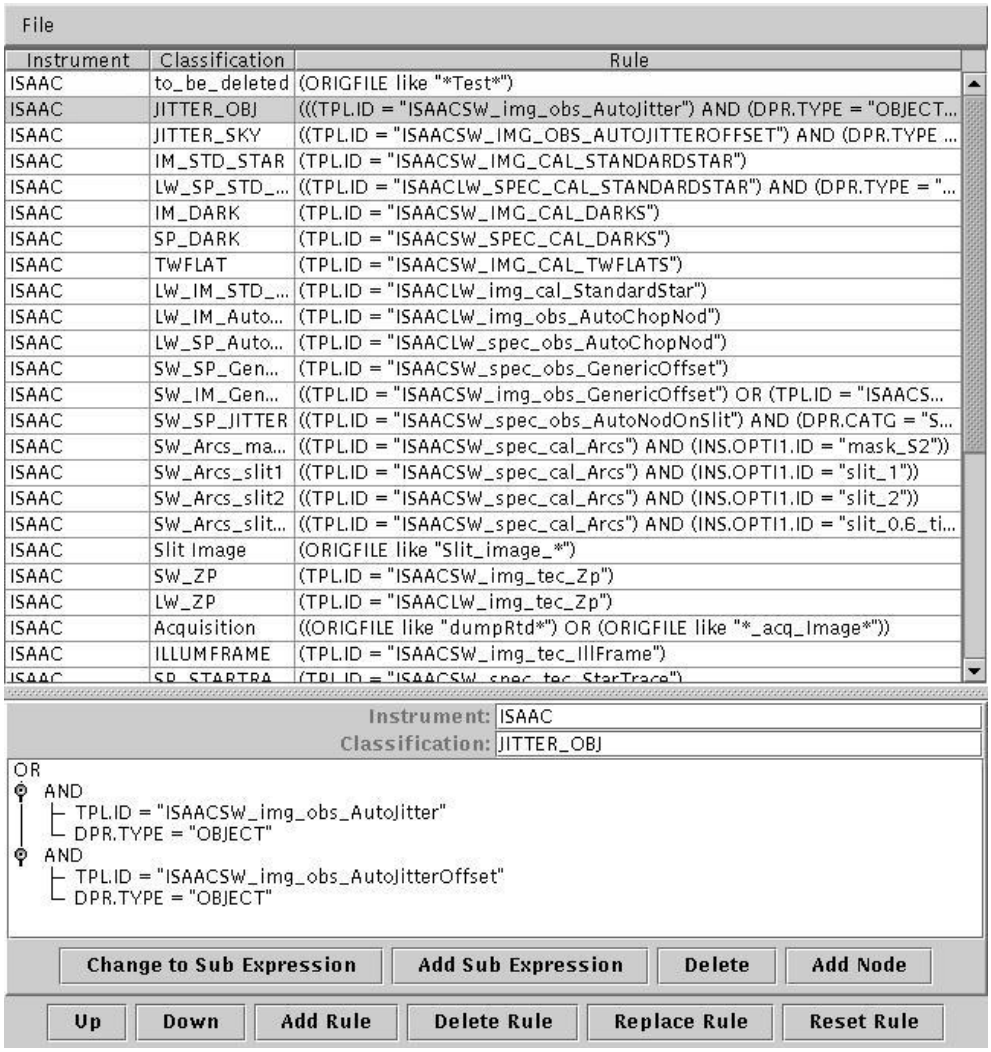
File
Instrument
Classification
Rule
ISAAC to_be_deleted (ORIGFILE like "*Test*")
ISAAC JITTER_OBJ (((TPL.ID = "ISAACSW_img_obs_AutoJitter") AND (DPR.TYPE = "OBJECT...
ISAAC JITTER_SKY ((TPL.ID = "ISAACSW_IMG_OBS_AUTOJITTEROFFSET") AND (DPR.TYPE ...
ISAAC IM_STD_STAR (TPL.ID = "ISAACSW_IMG_CAL_STANDARDSTAR")
ISAAC LW_SP_STD_... ((TPL.ID = "ISAACLW_SPEC_CAL_STANDARDSTAR") AND (DPR.TYPE = "...
ISAAC IM_DARK (TPL.ID = "ISAACSW_IMG_CAL_DARKS")
ISAAC SP_DARK (TPL.ID = "ISAACSW_SPEC_CAL_DARKS")
ISAAC TWFLAT (TPL.ID = "ISAACSW_IMG_CAL_TWFLATS")
ISAAC LW_IM_STD_... (TPL.ID = "ISAACLW_img_cal_StandardStar")
ISAAC LW_IM_Auto... (TPL.ID = "ISAACLW_img_obs_AutoChopNod")
ISAAC LW_SP_Auto... (TPL.ID = "ISAACLW_spec_obs_AutoChopNod")
ISAAC SW_SP_Gen... (TPL.ID = "ISAACSW_spec_obs_GenericOffset")
ISAAC SW_IM_Gen... ((TPL.ID = "ISAACSW_img_obs_GenericOffset") OR (TPL.ID = "ISAACS...
ISAAC SW_SP_JITTER ((TPL.ID = "ISAACSW_spec_obs_AutoNodOnSlit") AND (DPR.CATG = "S...
ISAAC SW_Arcs_ma... ((TPL.ID = "ISAACSW_spec_cal_Arcs") AND (INS.OPTI1.ID = "mask_S2"))
ISAAC SW_Arcs_slit1 ((TPL.ID = "ISAACSW_spec_cal_Arcs") AND (INS.OPTI1.ID = "slit_1"))
ISAAC SW_Arcs_slit2 ((TPL.ID = "ISAACSW_spec_cal_Arcs") AND (INS.OPTI1.ID = "slit_2"))
ISAAC SW_Arcs_slit... ((TPL.ID = "ISAACSW_spec_cal_Arcs") AND (INS.OPTI1.ID = "slit_0.6_ti...
ISAAC Slit Image (ORIGFILE like "Slit_image_*")
ISAAC SW_ZP (TPL.ID = "ISAACSW_img_tec_Zp")
ISAAC LW_ZP (TPL.ID = "ISAACLW_img_tec_Zp")
ISAAC Acquisition ((ORIGFILE like "dumpRtd*") OR (ORIGFILE like "*_acq_Image*"))
ISAAC ILLUMFRAME (TPL.ID = "ISAACSW_img_tec_IllFrame")
Instrument: ISAAC
Classification: JITTER_OBJ
OR
AND
TPL.ID = "ISAACSW_img_obs_AutoJitter"
DPR.TYPE = "OBJECT"
AND
TPL.ID = "ISAACSW_img_obs_AutoJitterOffset"
DPR.TYPE = "OBJECT"
Change to Sub Expression
Add Sub Expression
Delete
Add Node
Up
Down
Add Rule
Delete Rule
Replace Rule
Reset Rule

Figure 7: The **Classification Rules Builder** Panel. Some of the rules for the ISAAC instrument are displayed in the upper panel. The lower panel shows the classification rule for files of the type JITTER_OBJ. This rules is explained in more detail in the following text.

2) Double condition rule. In this case a double condition on one keyword or on two keywords is requred to classify a file:

```
AND
 ├─ TPL.ID="ISAACLW_IMG_CAL_DARKS"
 └─ DET.MODE.NAME="UncorrHighBias"
```

which classifies an ISAAC file as LW_Uncorrelated_Dark, whenever both conditions are satisfied. Note that AND is referred to as a **Node** and TPL.ID="ISAACLW_IMG_CAL_DARKS" is called a **Sub Expression.**

```
OR
 ├─ ORIGFILE like "dumpRtd*"
 └─
```

```
ORIGFILE like "*_acq_Image*"
```

which classifies a file as Acquisition image whether its ORIGFILE (Original filename) keyword contains the string dumpRtd* or the string *_acq_Image*.

3) Triple (multiple) condition rule. The simplest example for this rule is to check if three (or more) conditions are all meet or one of three (or more conditions) condition is met:

```
AND
├─ DPR.CATG="CALIB"
├─ DET.CHIPS=1.0
└─ DPR.TYPE="DARK"
```

which classifies a file a UVES files as a DARK_BLUE.

```
OR
├─ OBS.PROG.ID like "60.A-9023*"
├─ EXPTIME < 60
└─ INS.FILT1.ID != "V_BESS"
```

which classifies a file as FORS1 frame that was either observed under the engineering standard Run ID or its exposure time is less than 1 minute or its filter is not a V_BESS (this example is not in the distributed version of the classification rules).

4) Complex rules. These rules include combinations of nodes (AND and OR) and multiple sub expressions (conditions on keywords). The indentation gives the order of execution of each single block of constraints.

```
AND
├─ DPR.CATG="CALIB"
├─ AND
│   ├─ DPR.TYPE like "*WAVE*"
└─ │  └─ DPR.TYPE like "*LAMP*"
   DPR.TECH="SPECTRUM"
```

which classifies a frame as ARC_LSS for FORS2. The inline translation of this rule in "algebric" notation is:

```
(DPR.CATG="CALIB") & ((DPR.TYPE like "*WAVE*")&(DPR.TYPE like "*LAMP*"))&(DPR.TECH="SPECTRUM")
```

This rule can be simplified eliminating the innermost AND (&) clause and putting all 4 constraints under the same AND node.

```
AND
```

```
├─DPR.CATG="SCIENCE"
├─DPR.TYPE="SKY"
├─DPR.TECH="MOS"
└─OR
  ├─TPL.ID="VIMOS_mos_obs_Stare"
  └─TPL.ID="VIMOS_mos_obs_jitter"
```

which classifies as MOS_SCIENCE frames taken with VIMOS. The corresponding expression in "algebric" notation is:

```
(DPR.CATG="SCIENCE")&(DPR.TYPE="SKY")&(DPR.TECH="MOS")&((TPL.ID="VIMOS_mos
_obs_Stare")OR (TPL.ID="VIMOS_mos_obs_jitter")).
```

```
OR
├─AND
│  ├─ TPL.ID="ISAACSW_img_obs_AutoJitter
└─ └─ DPR.TYPE="OBJECT"
  AND
   ├─ TPL.ID="ISAACSW_img_obs_AutoJitterOffset
   └─ DPR.TYPE="OBJECT"
```

which classifies as JITTER_OBJ frames taken with ISAAC. The corresponding expression in "algebraic" notation is:

```
((TPL.ID="ISAACSW_img_obs_AutoJitter)&(DPR.TYPE="OBJECT")) OR
((TPL.ID="ISAACSW_img_obs_AutoJitterOffset)&(DPR.TYPE="OBJECT")
```

The same classification rule can be written in different ways. As a rule it is recommended to privilege first human readability, like in the case above, and then to use the logical expression with the least possible number of constraints. The higher this number, the slower will be the tool in classifying files.

- It is the responsibility of the user to ensure that there are no contradicting rules or that no rule is a subset of another. The tool applies the rules for file classification in the order in which they are present in the classification rules file. Therefore, the first rule is always applied.

- You may need to use a rule, that is a subset of another one. For example:

```
AND
├─TPL.ID="ISAACLW_IMG_OBS_AUTOJITTER"
└─DPR.TYPE="OBJECT"
```

and

```
AND
├─TPL.ID="ISAACLW_IMG_OBS_AUTOJITTER"
├─DPR.TYPE="OBJECT"
└─SEQ.CUMOFFSETX = 0
```

If this order is maintained on the classification rule file, the second rule will never be applied. Inverting the order of the rules ensures that both will be applied. As a rule, always put the most specific or restrictive rule on top.

4.2.1 Changing the order of the rules in the classification rules file.

To change the order of the rules in the classification rules file, use the two **Up** and **Down** buttons on the bottom of the window: first select the rule that has to be moved and then click on the button so many time as many position changes are required.

You can skip the section 4.2.2 and 4.2.3.if you are a first time user!!

4.2.2 Editing an existing rule – Advanced users

The easiest way to configure a personal set of classification rules, is to modify the rules distributed with the tool: the file ALL_rules.rul contain rules for all instruments of the VLT (commissioned at the time of the release), while ISAAC_rules.rul, FORS1_rules.rul, FORS2_rules.rul, UVES_rules.rul and VIMOS_rules.rul contain the classification rules for the specific instruments. The two FORS instruments have the same set of rules, but since the tool applies the rules on the basis of the `INSTRUME` keyword, they have to be specified twice, either with FORS1 or with FORS2 as instrument label. The FORS_rules.rul contains rules for both instruments.

It is possible to append a classification rule files to another one. Click the **Menu File>Append other rule file** in the **Classification Rule Builder. The browse window opens. Browse and select the file, click on** Append**.**

Suppose you want to modify the following rule:

```
AND
 ├─ TPL.ID=″ISAACLW_IMG_CAL_DARKS″
 └─ DET.MODE.NAME=″UncorrHighBias″
```

into

```
OR
 ├─ TPL.ID like ″*Jitter*″
 └─ TPL.ID notlike ″*Generic*″
```

1) To edit the node `AND`, double click on it (alternatively single click and then click on the leftmost button EDIT). A window with a toggle button which allows to select between the option `AND/OR`. Select `OR`. Confirm with `OK`.

2) Double click on the first term of the logical expression. A window appears (see Figure 8). On the top part there are two editable fields and in between them a toggle button. The first field contains the keyword, as seen on the header, without the HIERARCH.ESO part, for ESO standard fits keywords. Type in it `TPL.ID`. The syntax for keywords is case sensitive and requires dots in between strings.

3) Select with the toggle button the filed “`like`”

4) Type in in the leftmost field: `″*Jitter*″`

5) Confirm with `OK`.

6) Repeat steps 2) –5) to edit the second sub expression.

Whenever a syntactically wrong value is inserted in the field and ***OK*** is pressed, the content of the filed is reset to the previous value.

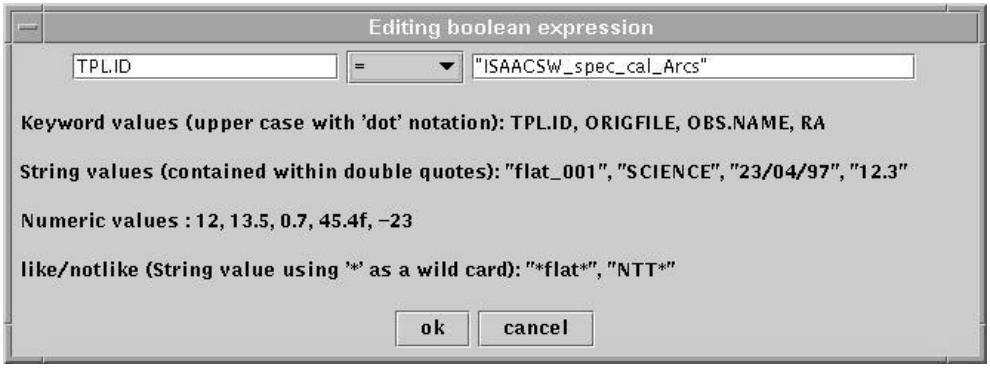
Editing boolean expression
TPL.ID
=
"ISAACSW_spec_cal_Arcs"
Keyword values (upper case with 'dot' notation): TPL.ID, ORIGFILE, OBS.NAME, RA
String values (contained within double quotes): "flat_001", "SCIENCE", "23/04/97", "12.3"
Numeric values : 12, 13.5, 0.7, 45.4f, -23
like/notlike (String value using '*' as a wild card): "*flat*", "NTT*"
ok
cancel

Figure 8: **Editing Boolean expression** window. Syntax and examples are explained in the text below the editable fields.

Below the editable field in **Editing Boolean expression** window, there are some basic examples on how to type the rules. As a basic rule, beware of always putting double quotes for strings, use `*` as wildcard characters (works only with the logical expression `like` and `notlike`, do not use quotes for numeric values.

It is possible to add a node after a sub expression, simply selecting the sub expression and clicking on the **Add Node** button. Viceversa, it is possible to add a sub expression under a node, clicking on the node and then on the **Add Sub Expression** button. Whenever an operation is not possible (e.g. adding a node under another node) the corresponding button his greyed out (e.g. **Add Node**). It is possible to transform a node into a sub expression, selecting the node and then clicking on the **Change to Sub Expression** button. In this event all sub expressions under this node, but the first one, are deleted.

Once the rule is modified, you submit it to the file, using the row of buttons at the bottom of the window (respectively the buttons are **Up, Down, Add Rule, Replace Rule, Reset Rule, Dismiss**). There are various possibilities:

1) Click on the **Replace Rule** button, to replace the existing rule with the new one

2) Click on the **Add Rule** button to add this rule to the list of existing one. Beware**:** to avoid having two rules with the same name, the **Classification field must be edited!!!**

3) Click on the **Reset Rule** button, to revert to the original rule (that is UNDO!!)

Before adding or replacing the rule, be sure of checking that the **Instrument** field is correct.

- If you add a rule before changing its classification, or, in other words, you have two rules with the same name, you can select the wrong one and delete it via the **Delete Rule** button. Alternatively you can select it, modify the **Classification** field and then use the **Replace Rule** button

- What happens if there are two identical rules or two rules with the same name? Nothing bad! The tool classifies the files using the rules in the order they are present in

the classification rule file. The second instance of the rule will therefore never be applied.

4.2.3 Creating a new rule – Advanced users

If the application is started with no classification rules file (a window with a warning will pop up at start up) the **Classification Rule Builder** window starts empty. To create a new rule, follow the step by step sequence:

1) Fill in the **Instrument** field. E.g.: `ISAAC`

2) Fill in the **Classification** field. E.g.: `SCIENCE_imaging`

3) Click on the rule editor panel, on the `KEYWORD="value"` string appear. Either

   a. Click on the **Edit** button and edit the rule (***double click*** of the string is also possible).

   or

   b. Click on the **Add Node** button.

   Case a. is to edit a single constraint rule, while option b. adds a node with two sub expressions.

   To edit and modify the nodes and/or the sub expressions proceed as told in the previous section.

4) Click on the **Add Rule** button.

Repeat steps 1)-4) till all the rules have been added.

4.2.4 Closing the Classification Rule Builder window

Once all the rules have been modified or created click on the **Dismiss** button, which will automatically save all changes in the rule file (which is the one is specified in the **Preference, Data** register).

Once the window has been closed, the changes take effect only if a "reclassify" is forced. To do so, click on **File, Reclassify in the** Menu bar**.**

4.3 The .gasganorc file

The .gasganorc file is created in the $HOME directory the first time the tool is started. The first version of this file contains default settings, which the user can modify as explained in the previous sections.

Every time the set up of the tool is modified, by means of the **Preference** panel, the corresponding setting is saved into the .gasganorc file. This is a standard ASCII file, containing a list of entries, with the format:

`KEYWORD=value`. E.g. `CLASSRULE_FILE=/my_gasgano_path/UVES_rules.rul` specifies the name of the classification rule file.

A list of all the possible entries in the .gasganorc file and their meaning is given in chapter 9.

It is recommended for the normal user to modify the preferences from within the tool, instead of editing manually the .gasganorc file

.

5 *A SIMPLE SESSION*

This chapter will describe a simple session with *Gasgano* and its most basic functionality. If you are a first time user please take some time to read this part. Start up procedures are described in Chapter 3.

5.1 Loading files into Gasgano

To load files into the application, click on **File>Add/Remove Files…**. A window appears (see Figure 9). The main panel of the window is empty when no directories or files are loaded.

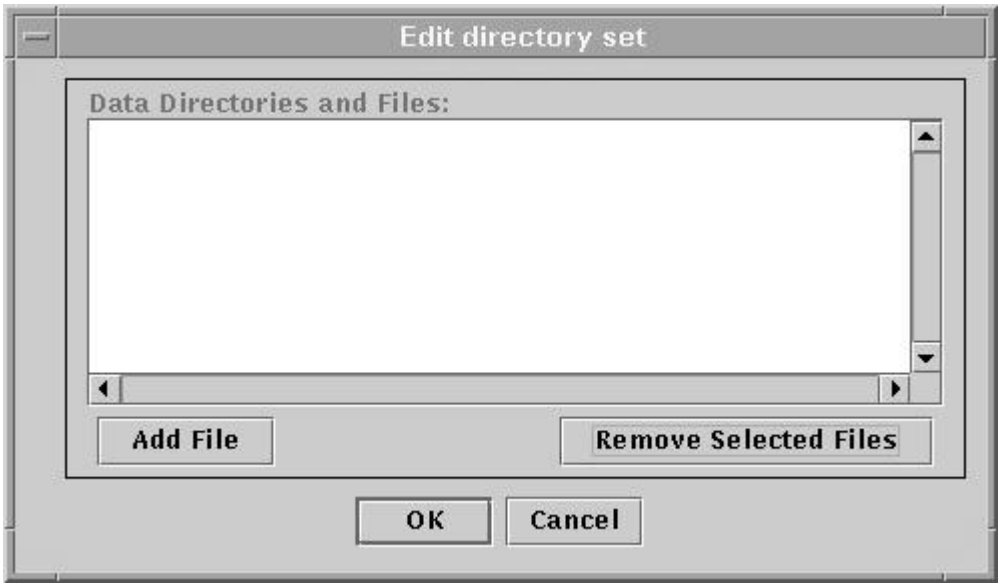

Figure 9: the **Edit directory set** window allows adding or removing directories/files to/from the **Navigation Panel.** In this example no files are loaded. Clicking on **Add File** starts another panel to browse in the file system.

To add directories and files, click on the **Add File** button. A new window appears, as shown in Figure 10.

The first time the panel is opened, the content of the `${HOME}` directory is shown. Subsequently the application will "remember" the last selected directory. It is possible to navigate through the file system either double clicking on a folder to change directory or using the first button after the combo box on the upper toolbar to move up one level. To select a directory to load (its name should appear in the field **File Name:** at the bottom) single click on its name and then press the **Add File** button. Multiple selections are NOT possible when adding directories. It is also possible to select a single file and add it the same way.

Loading a directory will automatically load all its subdirectories. If some of them are not needed, they have to be selected in the **Edit directory set** panel (Figure 9) and removed by clicking on the **Remove Selected Files** button**.** When removing directories or files from the list of loaded ones, multiple selection is possible. To learn how to do multiple selection, go to section 5.2.

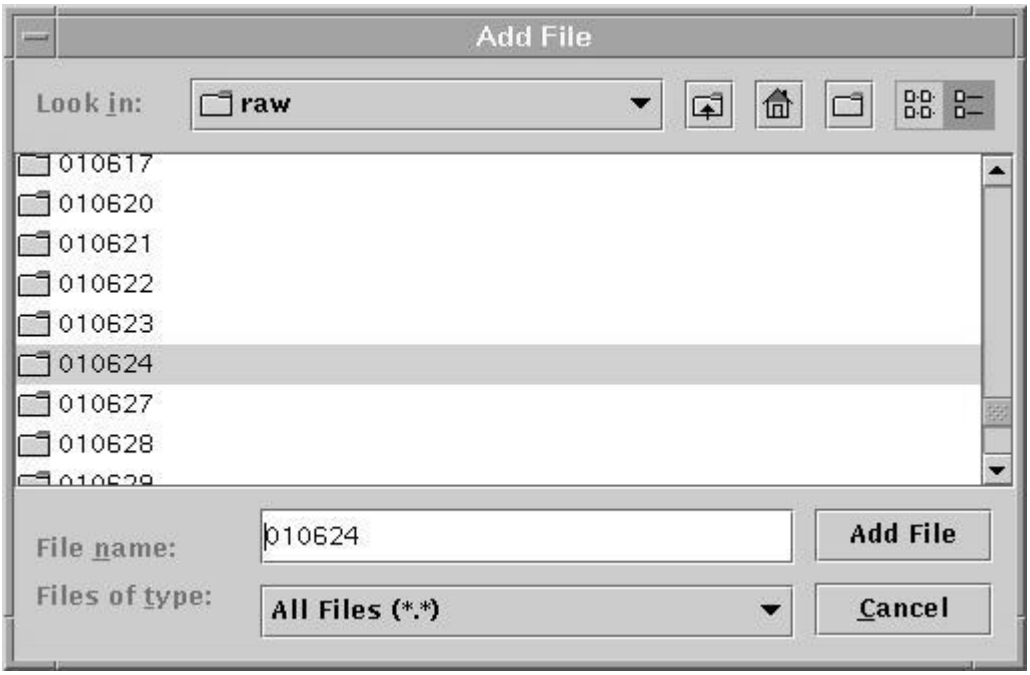

Figure 10: The **Add File** Panel allows browsing in the file system to select directories and Files for loading into the application. To select a directory/file, single click on it and then click on the **Add File** button. Multiple selections when adding files are **NOT** possible.

Once files are loaded by the tool, the **Navigation Panel,** the **File Association Panel** and the **File Details Panel**, will look like shown in Figure 2.

5.2 Changing the view

The view in the **Navigation Panel** can be changed in various ways, according to the user's preferences or needs.

5.2.1 Grouping files

On the **Toolbar** of the **Navigation Panel**, right of the icon buttons, there is a selector menu (combo box), which by default displays the **Default Grouping** field. The other available options are **Group by Directory** and **Group by Telescope**. The default setting leaves the tree unchanged and, as described in section 2.2, displays files grouped by their OBS.PROG.ID, the OBS.ID and the optional user's defined keywords. The **Group by Directory** option adds an extra layer on top of the default view and groups file by their location in the file system. The Complete absolute path of each group of files in the same directory is displayed in the panel, as shown in Figure 11.

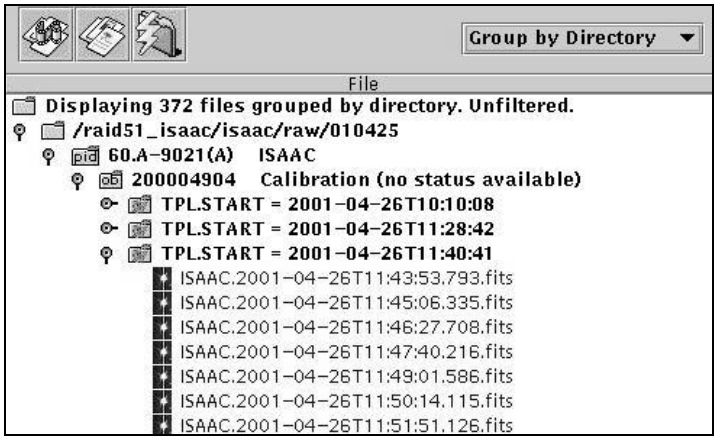

Figure 11: snapshot of the **Navigation Panel** when the **Group by Directory** option is selected. In addition to the three levels OBS.PROG.ID (60.A-9021(A)), OBS.ID (200004904) and TPL.START, the path of the files is included on top: these files are under the directory **/raid51_isaac/isaac/raw/010425**.

The **Group by Telescope** option works in the same way as the **Group by Directory** and adds a top layer which group files according to the keyword `TELESCOP` (e.g. `ESO-VLT-U1`). This option is mainly thought for ESO internal use.

5.2.2 Expanding and collapsing the view

The first time a data set is loaded into the application, the tree in the selected view (e.g. the **Default grouping**) is collapsed, and only the list of the top folders (e.g. the list of `OBS.PROG.ID`) is shown. In order to view the content of the folders below, single click on any point on the folder line (i.e. a click on the node, the folder icon or on the name). Clicking a second time, closes the folder (that is, the double click has no effect, and if you are fast enough no changes in the view are noticeable!!). Clicking through the various layers of the tree will expand/collapse it. It is possible to expand or collapse the entire tree at once, by means of the **Expand/Collapse** toggle button on the **Toolbar**.

The **Expand/Collapse** is a "toggle" button in the sense that when the tree is collapsed (such as after a data set is loaded) the button displays "Expand". When it is expanded, it displays "**Collapse**". Sometimes the button shows the "wrong" value. It may be set to "**Expand"** even if the tree is partially expanded or vice versa. This is because while the user has the option to click on one or more folders to expand them, the tool (and the **Expand/Collapse** button) has got no memory of this action. Therefore it does not change the status (value) of the button. Clicking twice on it forces a reset.

5.2.3 Introducing an extra level in the tree

Figure 11 shows a tree with four layers or levels. The **Group by Directory adds the first layer** and it is optional. The second and third are by default (and usually are the first and second layers in the **Default grouping** view). The fourth level is optional and can be configured by the user, as explained in section 4.1.2. Whenever a keyword is chosen for the additional grouping, all files with an header (that is fits and tfits), which share the same value for that keyword are grouped together, provided they already share the same higher level grouping. In Figure 11, the extra level is defined by the keyword `TPL.START`, which is set to the same values for all frames taken within the same run of an observing template. All frames share the Observations Block (`OBS.ID`)

for a defined run (`OBS.PROG.ID`) and are located in the same directory in the file system. Any keyword can be used to define the extra layer, provided it is present in all the headers. If a file does not contain the selected keyword, it will be placed in the same `OBS.PROG.ID` and `OBS.ID` folders, at the same level in the tree as the additional layer.

The application lists the keyword used for the optional grouping and its value on the folder line in the **Navigation Panel** (in the figure the files with `TPL.START=2001-04-26T11:40:41` are expanded).

5.2.4 Resizing the fields

All fields in the **Navigation Panel** can be resized to accommodate the length of the keywords' values. The fields names are displayed in a bar just below the main **Toolbar,** in the order specified in the **Preferences Display and Printing** register. When the mouse pointer is on the separation line between two fields, it changes shape. Clicking the mouse left button allows grabbing the separation line and resizing the field to the preferred length.

5.3 Selecting Files

Any operation on file(s), such as viewing the header content, displaying a frame with a viewer, copying/moving, creating reports, etc, requires a selection. Selected files appear highlighted in light blue. Single selection is done by single clicking on the object (a file, a directory, a folder). Multiple selection is possible in two ways:

After selecting the first file via a single click, the user can:

1) press the ***Shift*** key while mouse clicking on another file: this selects a block of contiguous files;

2) press the ***Ctrl*** key while mouse clicking on another file: this adds individual files to the selected list.

Other options for file selection, include:

1) Click on any node in the tree with the right mouse button: this expands the node and the sub-nodes below it and selects all the files within.

2) Click on any already expanded node with the right mouse button: this selects all files within that node.

Press the ***Ctrl*** key while right mouse clicking on a node will allow multiple selections of nodes

In all cases, if files are selected, the action of clicking will deselect the file(s).

In Section 8.1,

Table 2 summarises all mouse click options.

- ***Right clicking*** on the uppermost folder icon in the **Navigation Panel** (e.g. in Figure 2 the one with the `Displaying 98 files. Unfiltered` **folder line**), selects all the files loaded by the application. To deselect them, either *left-click* on a single file or use the **Expand/Collapse** button on the **Toolbar.**

5.4 Filtering files

Gasgano offers the possibility to filter files loaded in the **Navigation Panel**, on the basis of keyword properties or values. The concept is similar to that of the classification rules. It is possible to set a constraint on a keyword, or a set of keywords by means of `AND/OR` logical expressions, and to have the tool displaying only those files that fulfil the condition.

To filter files click on the **Menu bar** under **File> Filter files…**. A new panel, called **Filter Builder**, appears (Figure 12). It consists of an upper window, empty at first use. After each use of the filtering option, the filter rules will be remembered and listed in this window for re-use. The lower window allows editing in the same way as the **Classification rules builder** window. Refer to section 4.2, for a description of the syntax and how to edit the rules. The rule displayed in the lower window of the **Filter Builder** is the one that is applied to the loaded data set, once the **OK** button is pressed. Rules are saved in the `${HOME}/gasgano/filter.rul` file, unless otherwise specified in the **Preferences**.

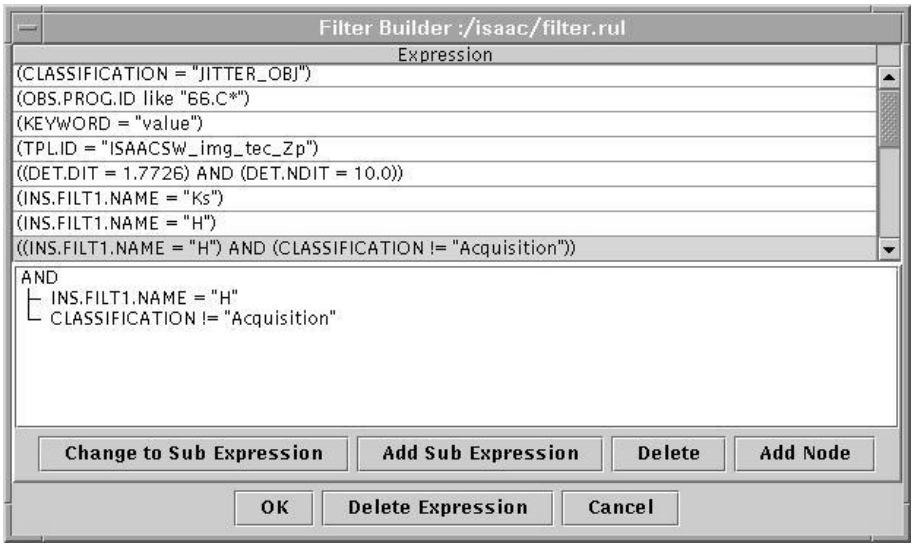

Figure 12: The **Filter Builder** window. In this example, rules are saved in the file **filter.rul.** It is possible to configure the file and its location through the **File>Preferences** panel. The upper window shows a list of rules. The selected rule is displayed in the lower window, and it is the one which is applied to the data set loaded by the application: all files with filter set to "H" and with CLASSIFICATION other than "Acquisition" are displayed. The others are "filtered out" by the application.

Once the filter has been applied, the first row of the **Navigation Panel,** displays a message**.** In the example of Figure 12, the message is:

```
Displaying X files. Filter: ((INS.FILT1.NAME=”H”) AND
(CLASSIFICATION!=”Acquisition”))
```

Where "X" is equal to an integer number (including 0 if no files that fulfil the constraints are found). To remove the filter click on the **Menu bar** the **File>Unfilter files**.

- When a data set is filtered, the application keeps in memory the files that have been filtered out. Therefore, removing a filter does not require files to be reloaded from disk.

- The filter files option is very useful when dealing with a large data set. It is recommended to load the entire data set the first time and then to apply filters. One example, is for instance to filter out all calibration files (DPR.CATG!="CALIB") when just a quick look of the scientific frames is needed.

5.5 Reclassify and/or Refresh the view

It is possible to force the application to reclassify all files loaded (that is, re-apply the classification rules) clicking in the **Menu Bar** on **File>Reclassify**. Reclassification of files is done automatically by the application every time the classification rule file name is changed in the **Preferences**, but it has to be done manually by the user every time (s)he modifies the rules.

The **Refresh option,** under **File>Refresh** in the **Menu bar,** allows refreshing the view in the **Navigation Panel**. The automatic refresh option is useful when the content of the loaded directories changes frequently (for instance while moving or copying files outside the tool). If this is not the case, since the application freezes for a short time during the refresh, it is recommended to set the field in the **Preferences** to **No update** and perform a manual refresh in case is needed.

- It is possible to have an automatic refresh of the loaded directories by setting up the refresh time in the **File>Preferences… Data** register.

5.6 Finding tools

It is possible to find entries in the **Navigation Panel** and in the **File Details Panel** by means of the **Find tools**. To find an entry in the **Navigation Panel**, type the string to be looked form in the field **Find an entry** on the upper **toolbar** and then click on **find**, to the right. Clicking twice or more on it will search for other entries of the same string in the panel. The string is searched "as is" and without wildcard characters must be entered. The search is case insensitive. Once found in the panel the line containing the string is highlighted.

To find an entry in a fits header in the **File Details panel** proceed in the same way, using the **Find in header** filed on the panel's toolbar.

5.7 Displaying Files

Gasgano offers the possibility to define default file viewers, as explained in the section 4.1.1, even though NO viewers are distributed together with the application. The simplest way to display a file is to select it and to click on the leftmost button icon on

the **Toolbar**. The selected file is passed to the defined application/script, which displays the file. It is possible to define three types of viewers, which will automatically started by the tool, according to the type of selection in the **Navigation Panel**:

1) Single tfits table: the tool runs the tfits viewer

2) Single fits selection: the tool runs the fits file viewer

3) Multiple fits file selection: the tool runs a viewer that accepts data cubes in input.

As explained in section 4.1.1, it is advisable to call to a script rather than a viewer application, that the viewer may be run in the background. This avoids freezing *Gasgano*. The example script, load_rtd, given in the same section, shows how it is possible to configure a viewer in such a way that it automatically takes input files from *Gasgano*.

The recommended RTD viewer is also capable of dealing with data cubes.

In addition to the tool bar button, the viewer can be launched to display a file selection through the **Menu Bar,** Files>Display… **item.**

5.8 Copying/Moving Files

Gasgano can move or copy files from their loading location to another selected by the user. After selection of file(s), click on the **Menu Bar** on **Selected Files>Copy** or **Selected Files>Move.** A window similar to that shown in Figure 10**,** appears: it allows browsing on the file system in order to select the destination directory. When the location has been selected, click on the **Copy/Move** button. According to the operation *Gasgano* may behave in different ways:

1) If files are moved to a new directory (i.e. not loaded), the directory is automatically loaded by the tool

2) If files are copied to a new directory the directory is not automatically loaded

3) If files are copied or moved to a directory, which is already loaded, nothing happens. Manual refresh has to be forced (click on **File>Refresh** in the **Menu bar**.)

It is not possible to delete a file with *Gasgano*, but you can move files to a "wastebasket" directory in your file system and then remove them manually.

5.9 Creating a tar file

To create a tar file out of a set of selected files click on the **Menu bar**, **Selected Files>Tar…. .** A window similar to the one shown in Figure 10**,** appears. Select the destination directory and type the name of the tar file in the field **File name,** at the bottom. Click on the **tar** button. The tool will "freeze" for the time needed to create the

tar file (that is the time needed by the underlying UNIX workstation to perform the job). This can be quite long if the number of files is large. The syntax of the tar command carried on by *Gasgano* is:

```
tar –cbf 20 - <list of parameters>
```

where `<list of parameters>` is the list of files selected in Gasgano.

5.10 Creating a keyword list file

Keyword lists are ASCII files used by *Gasgano* to filter file headers or to print reports, so that only the keywords present in the list are shown in the **File Details Panel** or printed.

To create a keyword list file from scratch with *Gasgano*, click on the **Menu Bar** on **Tools>Edit Keyword Lists….** A new window, called **Keyword List Editor appears.** It contains two columns, **labelled Keyword Name** and **Report Label**. At the bottom there are the buttons **Insert Row** and **Delete Row**. On the upper **Menu Bar**, the **File Menu** contains the entries: **Open, New, Save As** and **Quit.**

Add as many rows as needed with the **Insert Row** button. In the first column enter the fits keyword as it appears in the file header, without the HIERARCH ESO part, in uppercase and in "dot" notation. Figure 13 shows an example. The **Report Label** column allows the user to specify the label used by the application when creating a report. The syntax and the contents are free. Once the list is complete, click on **File>Save As…** to save the file. The file should be (optionally) saved in the directory specified in the **Preferences Data** register for the keyword lists, since this is the directory where *Gasgano* looks when trying to load a keyword list file.

To edit an existing file, click on **File>Open** in the **Menu bar** of the keyword list editor. The tool opens the usual file system browse window, in the default directory (as specified in the **Preferences**). Browse and/or select the file, click **Load**. Edit the file as needed and then save. To create a new file, click on **File>New**.

File>Quit quits the editor without saving the file.

The keyword list files are standard ASCII files and can be edited with any editor, without having to use the built in editor.

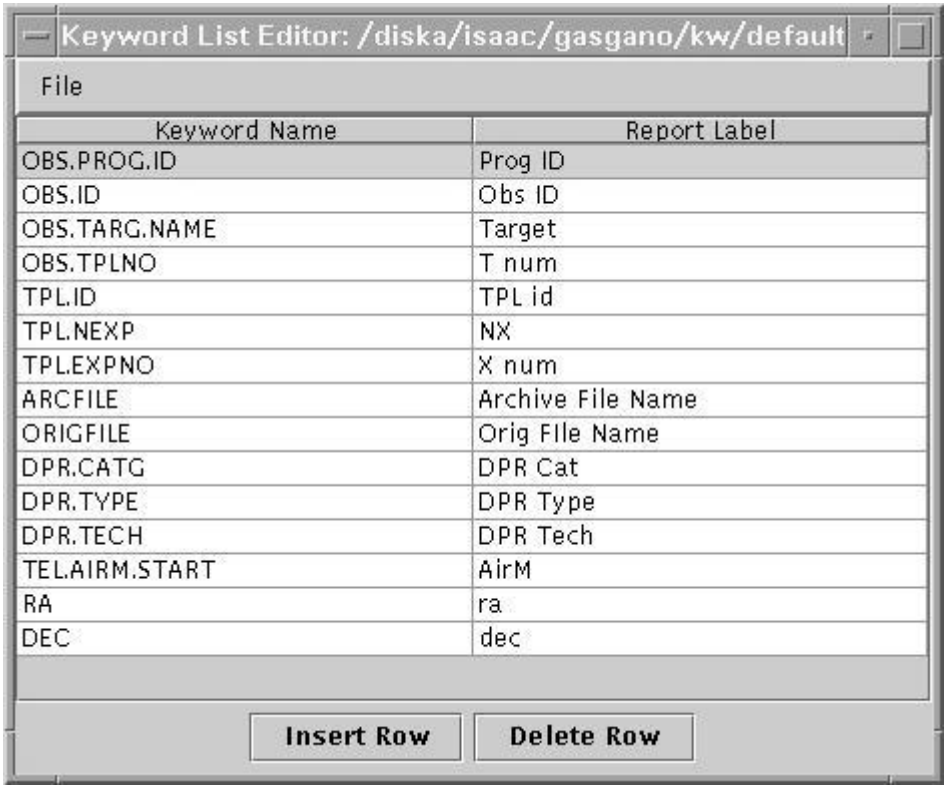

Figure 13: the **Keyword List Editor**. The list of keywords has been edited and the file saved to disk.

5.11 Filtering file headers

On the top part of the **File Detail Panel,** there are a **Load Filter** button and a **Filter** toggle on/off button. Click on Load filter to have the browsing window opening on the default directory for keyword lists. Select the file and click on **Load**. Click on the toggle button **Filter**, so that it is blackened. If a file in the **Navigation Panel** is selected, only the header keywords present in the keyword list file will then be displayed. To remove the filter, deselect the toggle button **Filter**. To load another keyword list file, repeat from the beginning.

When the fits header filtering is on, the **Find in header** works only on the currently displayed keywords.

5.12 Creating a report

A report is an ASCII file that displays file header keywords for a set of fits and/or tfits files in a tabular form. To create a report a keyword list file is needed. If none is available, create it as explained in section 5.10. Then:

1) Select fits or tfits files in the **Navigation Panel (as specified in section 5.3)**;

2) Click on the **Report** button on the main **Toolbar** (the second icon button from left) or click on **Selected files>Report…;**

3) The **Load Keyword list** window appears. At first use it is empty, while for subsequent uses the last loaded keyword list is kept in memory;

4) If it is empty, click on the **Load Keyword List** button at the bottom; the browse window opens in the default directories for keyword list files. Browse and select a file.

5) Modify the loaded file if needed, using the **Insert Row** and **Delete Row** buttons. To add a row above another one, select reference line and right click on the **Insert Row** button. Edit the fields as needed.

6) Click on **OK**.

7) The report is generated and the report window is opened (see Figure 14).

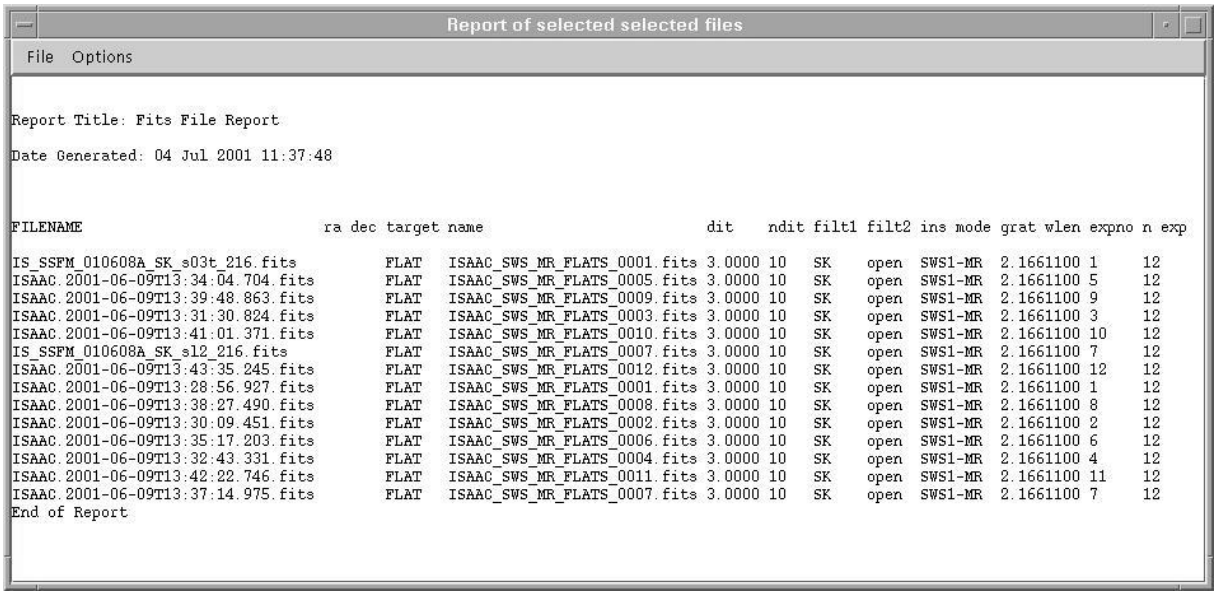
Report of selected selected files
File Options
Report Title: Fits File Report
Date Generated: 04 Jul 2001 11:37:48
FILENAME ra dec target name dit ndit filt1 filt2 ins mode grat wlen expno n exp
IS_SSFM_010608A_SK_s03t_216.fits FLAT ISAAC_SWS_MR_FLATS_0001.fits 3.0000 10 SK open SWS1-MR 2.1661100 1 12
ISAAC.2001-06-09T13:34:04.704.fits FLAT ISAAC_SWS_MR_FLATS_0005.fits 3.0000 10 SK open SWS1-MR 2.1661100 5 12
ISAAC.2001-06-09T13:39:48.863.fits FLAT ISAAC_SWS_MR_FLATS_0009.fits 3.0000 10 SK open SWS1-MR 2.1661100 9 12
ISAAC.2001-06-09T13:31:30.824.fits FLAT ISAAC_SWS_MR_FLATS_0003.fits 3.0000 10 SK open SWS1-MR 2.1661100 3 12
ISAAC.2001-06-09T13:41:01.371.fits FLAT ISAAC_SWS_MR_FLATS_0010.fits 3.0000 10 SK open SWS1-MR 2.1661100 10 12
IS_SSFM_010608A_SK_s12_216.fits FLAT ISAAC_SWS_MR_FLATS_0007.fits 3.0000 10 SK open SWS1-MR 2.1661100 7 12
ISAAC.2001-06-09T13:43:35.245.fits FLAT ISAAC_SWS_MR_FLATS_0012.fits 3.0000 10 SK open SWS1-MR 2.1661100 12 12
ISAAC.2001-06-09T13:28:56.927.fits FLAT ISAAC_SWS_MR_FLATS_0001.fits 3.0000 10 SK open SWS1-MR 2.1661100 1 12
ISAAC.2001-06-09T13:38:27.490.fits FLAT ISAAC_SWS_MR_FLATS_0008.fits 3.0000 10 SK open SWS1-MR 2.1661100 8 12
ISAAC.2001-06-09T13:30:09.451.fits FLAT ISAAC_SWS_MR_FLATS_0002.fits 3.0000 10 SK open SWS1-MR 2.1661100 2 12
ISAAC.2001-06-09T13:35:17.203.fits FLAT ISAAC_SWS_MR_FLATS_0006.fits 3.0000 10 SK open SWS1-MR 2.1661100 6 12
ISAAC.2001-06-09T13:32:43.331.fits FLAT ISAAC_SWS_MR_FLATS_0004.fits 3.0000 10 SK open SWS1-MR 2.1661100 4 12
ISAAC.2001-06-09T13:42:22.746.fits FLAT ISAAC_SWS_MR_FLATS_0011.fits 3.0000 10 SK open SWS1-MR 2.1661100 11 12
ISAAC.2001-06-09T13:37:14.975.fits FLAT ISAAC_SWS_MR_FLATS_0007.fits 3.0000 10 SK open SWS1-MR 2.1661100 7 12
End of Report

Figure 14: the **Report** window. A list of files (previously selected in the **Navigation Panel**) is displayed with their keyword values. If a keyword is not present I the header a blank space is left (like for **ra** and **dec** in this example).

On the **Menu bar** of the **Report** window the **File** and **Options Menus** are available:

The **File Menu** contains the **Print, Save As…** and **Quit** entries. **Print** sends the report file to the default printer (as specified in the **Preferences, Display and Printing** register**);** Save a… allows saving the file to disk (the usual browse window is open).

The **Options Menu** contains the **Select Fonts** entry. When selected it opens the **Edit fonts** window, which allows selecting a different font type and size from the one selected in the **Preferences, Display and Printing** register**.** The selected font is used only for the current report. To save it permanently edit the entry in the **Preferences.**

Monospaced fonts are ideal for printing because they maintain the correct alignment of the columns. Other fonts may not do this.

6 *AN ADVANCED SESSION*

This chapter described advanced features of the application.

6.1 Running external executables

Just as a set of selected files can be sent to the tar, copy and move command, it may be also sent to a user selected executable. The files selected in the **Navigation Panel** are passed as command line parameters, one file per parameter, with the full path name. On demand, the full path may be dropped and the external command executed after a `cd' into the common directory of all selected files (This requires that all selected files are stored in the same directory.). In order to drop the path, activate "shorten data files paths for commands" button in the **File>Preferences Data** register.

There are two ways to send selected files to a custom program:

1. Select one data set and use the option **Run** from the **Selected Files** menu in the **Menu bar**. A browse window appears. Locate the script, select it and click on **Run**(or double click on the script name to execute). Click ***Ok***in the confirmation window.

2. Use the S**cript Board**. This option is suitable to a situation where several instrument dependent external commands have to be executed frequently.

6.2 The Script Board

The Script Board is an additional window, which shows a configured set of commands corresponding to executable files in the file system. The window can be left open while using *Gasgano*. It is activated by clicking on the **Script Board** button (the third from the left) on the **tool bar** or selecting **Script Board** from the **Tools** menu or typing Ctrl-s when the mouse pointer is on the main *Gasgano*'s window. To execute one command one has to double-click on it.

Figure 15 shows the Script Board.

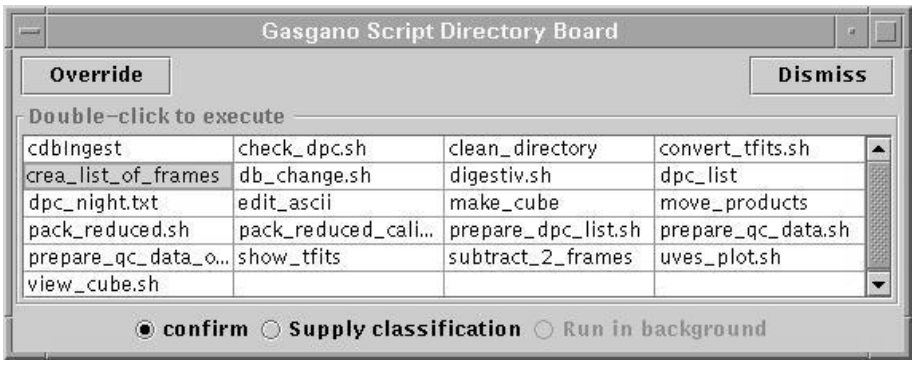

Figure 15: The **Script board.**

The **confirm** option will trigger a pop-up window prior to running the specified executable, which request an additional confirmation from the user.

On the Script Board the **Supply classification** option will pass the `CLASSIFICATION` Meta-keyword along with each file name. The sequence passed to the script is of the kind:

```
Filename1 CLASSIFICATION1 Filename2 CLASSIFICATION2 …
```

The **Run in background** option is disabled in the present version of *Gasgano*.

The **Script Board** automatically looks for the executables in a directory (the **common script** directory. This directory cannot be a link to another directory) specified using the **Preferences Editor** in the Data register. When the **Script** Board is open and a data set (one or more files) is selected, *Gasgano* checks if all files share the same `INSTRUME` keyword. If this is the case, the **Script Board** looks into an instrument specific directory, specified as a subdirectory of the script directory named after the `INSTRUME` keyword itself. A typical example situation looks as follows:

```
<gasgano-script-dir>/files common to all instruments (if any)
<gasgano-script-dir>/ISAAC/files for isaac....
<gasgano-script-dir>/FORS1/files for fors1....
<gasgano-script-dir>/EMMI/1.57/files for EMMI release 1.57....,
```

where `<gasgano-script-dir>` is the **common script** directory set in the **Preferences**.

If the selected files do not share a common `INSTRUME` keyword *Gasgano* shows the files in the `<gasgano-script-dir>` directory. The **Override** button on the top left corner of the **Script Board**, overrides the instrument specific settings and repositions the view of the **Script Board** in the **common script** directory.

6.3 Passing a directory path to an executable

It is possible to pass the path of a directory (loaded into the application) on the file system to an executable. To do this:

1) Select the **Group by directory** view on the **Tool bar.** In the **Navigation** Panel a collapsed list of folders with the directories names (paths) should be visible;

2) Click on the desired directory with the **middle** button of the mouse. A beep sound should be heard. This way *Gasgano* has saved into an internal clipboard the directory name.

3) Double click on the desired script in the **Script Board.** If the **confirm** option on the board is active the confirm window should give the message:

```
/{gas-script-dir/../your_script /path_of_the _selected directory
```

Clicking with the middle mouse button on the `OBS.PROG.ID` or `OBS.ID` folders in the **Navigation Panel** has the effect of saving the content of the folder line into the internal clipboard. Since the tool does not make a check on the meaning of the line but passes it “as is” to the executable, it is responsibility of the user to adapt the script according to the way *Gasgano* passes arguments. If for

example, the user clicks with the middle button on the `OBS.PROG.ID` folder containing data for a run ID, the folder line passed to the executable is of the kind: `OBS.PROG.ID INSTRUME (e.g. 60.A-9021(A) ISAAC)`

6.4 Example scripts

Gasgano can run executables, in the sense that it passes a list of parameters (the files currently selected in the **Navigation Panel**) to an external application. The executable may be anything (a shell script, a C program, etc) that can be run from the command line. It is responsibility of the user to create executables that handle the list of parameters (file names and/or classification string) in a correct way.

In order for *Gasgano* to "see" the application and to display it in the **Script Board,** the executable or a link to it must be put in either the **common script** directory or in one of the instrument specific directories.

In all the following examples of scripts, substitute `mypath` with the appropriate path to the executable in your file system. For those examples which use the eclipse library, see references [5] .

Examples:

1. Parameters passed to an executable: when files are selected and passed to an executable *Gasgano* may sends three different outputs. Consider the script that saves the list of parameters sent by *Gasgano* to the file /tmp/output.txt

```
##!/bin/sh
#
# Example shell script to be used with Gasgano:
# Print output – save to output.txt the list of
parameters sent #             to an executable.
#-------------------------------------------------
----------
echo $* > /tmp/output.txt
```

Assume that the following files are selected in the **Navigation Panel**:

```
ISAAC.2001-06-08T22:28:22.108.fits
ISAAC.2001-06-08T22:28:47.361.fits
ISAAC.2001-06-08T22:29:12.282.fits
ISAAC.2001-06-08T22:29:38.666.fits
ISAAC.2001-06-08T22:30:03.865.fits
ISAAC.2001-06-08T22:30:28.985.fits
ISAAC.2001-06-08T22:30:54.343.fits
```

1) In the default case files are passed as a list of names with full path separated by a space:

```
/diska/test/ISAAC.2001-06-08T22:28:22.108.fits /diska/test/ISAAC.2001-06-
08T22:28:47.361.fits /diska/test/ISAAC.2001-06-08T22:29:12.282.fits
/diska/test/ISAAC.2001-06-08T22:29:38.666.fits /diska/test/ISAAC.2001-06-
08T22:30:03.865.fits /diska/test/ISAAC.2001-06-08T22:30:28.985.fits
/diska/test/ISAAC.2001-06-08T22:30:54.343.fits
```

Even if the files are shown in the **Navigation Panel** without the path (the option **truncate filenames** in the **Preferences, Display** register is turned on), the path is passed to the executable

2) In case the **Supply Classification** toggle button on the **Script Board** window is turned on, each file and its classification separated by a space are passed as a list:

```
/diska/test/ISAAC.2001-06-08T22:28:22.108.fits TWFLAT /diska/test/ISAAC.2001-06-
08T22:28:47.361.fits TWFLAT /diska/test/ISAAC.2001-06-08T22:29:12.282.fits TWFLAT
/diska/test/ISAAC.2001-06-08T22:29:38.666.fits TWFLAT /diska/test/ISAAC.2001-06-
08T22:30:03.865.fits TWFLAT /diska/test/ISAAC.2001-06-08T22:30:28.985.fits TWFLAT
/diska/test/ISAAC.2001-06-08T22:30:54.343.fits TWFLAT
```

3) In case the **Supply Classificat**ion toggle button on the **Script Board** window is turned off and the **shorten data files paths for command** toggle button in **Preferences, Data** register is set off, files are passed as a list separated by spaces:

```
ISAAC.2001-06-08T22:28:22.108.fits ISAAC.2001-06-08T22:28:47.361.fits ISAAC.2001-06-
08T22:29:12.282.fits ISAAC.2001-06-08T22:29:38.666.fits ISAAC.2001-06-
08T22:30:03.865.fits ISAAC.2001-06-08T22:30:28.985.fits ISAAC.2001-06-
08T22:30:54.343.fits
```

2. Linking to an ASCII editor: to edit an ASCII file (for instance a reduction block file created form a virtual reduction block and saved to file, see section 6.5) create a link in the **common script** directory:

```
ln –s /mypath/emacs edit_ascii
```

The `edit_ascii` file will be visible in the **Script Board. To use** it select an ASCII file and double click on `edit_ascii`. NOTE: *Gasgano* freezes until the emacs session is concluded. The same procedure can be followed to create a link to any executable

3. A simple parser for the list of parameters: this shell script expects as input a list of file names from *Gasgano* (it assumes the option **Supply Classification in the** Script **is** turned off). The file should be saved in the **common script** directory (e.g. call it `crea_list_of_frames`). When executing, select the files in the **Navigation Panel** and double click on **crea_list_of_frames in the** Script Board**.** If the **Script Board** is positioned on an instrument specific script directory, use the **override** button (reference XXX).

```
#!/bin/sh
#
# Example shell script to be used with Gasgano:
# crea_list_of_frames.sh - creates a file
framelist.ascii to be used with eclipse recipes
#-------------------------------------------------
----------
#-------> move to the working directory
GAS_TMP_DIR=/mypath/gasgano
cd $GAS_TMP_DIR
```

```
#-------> Clean old stuff

if [ -s ${GAS_TMP_DIR}/framelist.ascii ]
then
        rm ${GAS_TMP_DIR}/framelist.ascii
fi

#-------> Format input for eclipse routine

list=`echo $*`

for i in $list
do
        printf "%s\n" $i >> framelist.ascii
done
```

4. Executing a data reduction recipe: this shell script passes a list of frames to the eclipse recipe that creates imaging flat fields. It should be saved in the directory <gasgano-script-dir>/ISAAC/…. In this example the file is called twflat. The typical input is a list of files selected in the **Navigation Panel**:

```
      /mypath/ISAAC.2001-07-05T22:18:35.944.fits
      /mypath/ISAAC.2001-07-05T22:18:52.351.fits
      /mypath/ISAAC.2001-07-05T22:19:08.761.fits
      /mypath/ISAAC.2001-07-05T22:19:25.173.fits
      /mypath/ISAAC.2001-07-05T22:19:41.581.fits
      /mypath/ISAAC.2001-07-05T22:19:57.992.fits
      /mypath/ISAAC.2001-07-05T22:20:12.537.fits
      /mypath/ISAAC.2001-07-05T22:20:28.947.fits
      /mypath/ISAAC.2001-07-05T22:20:45.355.fits
      /mypath/ISAAC.2001-07-05T22:21:01.763.fits
#!/bin/sh
#
# Example shell script to be used with Gasgano:
# twflat – starts the  eclipse is_twflat recipe
# requires the following eclipse routines:
# isaacp, dfits, fitsort, is_twflat
#-------------------------------------------------
----------
#-------> move to the working directory
GAS_TMP_DIR=/mypath/gasgano
cd ${GAS_TMP_DIR}
#-------> Define a unique name for files
name=`basename $1 .fits`

#-------> Clean old stuff

if [ -s /tmp/${name}.ascii ]
then
        rm /tmp/${name}.ascii
```

```
fi
#-------> Format input for eclipse routine

list=`echo $*`

for i in $list
do
        printf "%s\n" $i >>  /tmp/${name}.ascii
done

#-------> Run eclipse is_twflat

xterm -e sh -c "\\
     $E_PATH/bin/isaacp twflat -e -b -i
     /tmp/${name}.ascii -o ${GAS_TMP_DIR}/${name};
     \\
      $E_PATH/bin/is_rename ${GAS_TMP_DIR}/${name}*;
\\
      rm /tmp/${name}.ascii; \\
      read dummy" &
```

The part in bold in the example shows the call to the script. Here **$E_PATH/bin/** is the standard path to eclipse routines (once you have installed the software). In order to see the output of the eclipse routine xterm is called:

- The options **–e sh –c "…command1;command2;command3"** allow the user to execute a series of commands in the terminal session. For more information on the options for **xterm** see the Unix man pages.
- The first command **$E_PATH/bin/isaacp twflat** calls the eclipse routine with some options **(-e –b –e –o**, see the **twflat** online help manual page within isaacp [5]);
- the second command (separated by a ;) **$E_PATH/bin/is_rename** calls another eclipse routine, which renames the product **$name**, which is the output of the twflat recipe in **$GAS_TMP_DIR** , using the ESO naming convention for data products (online help: type **is_rename**). This command is optional and displayed here for explanatory purposes.
- The last command, **read dummy,** serves the purpose to leave the xterm session open and to check the output of the eclipse routine. Pressing the <***RET***> key in the window closes it.
- The **&** at the end provide a means to run the **xterm** and commands in the background, without "freezing" *Gasgano.*

6.5 File associations

At the end of the data reduction process, additional keywords are written in the header of the data products. These keywords (`HIERARCH.ESO.PRO.REC1.XXX.XXX`, where the XXX.XXX may vary) provide a means to associate raw files, their products and the master calibrations product used in the data reduction process. The **File Association Panel uses these keywords to show** associations between raw frames and frames reduced by any one of the instrument data reduction pipelines. As explained in section.2.2, *Gasgano* is able to reconstruct these associations and display them in a "virtual reduction block". When the user selects a file in the **Navigation Panel**, all the associations between this and other files are displayed in the **File Association Panel.** The associations displayed may list name of files that are not currently loaded in *Gasgano*.

Note that pipeline products produced earlier than period 68 (that is earlier than October 2001) may not contain the PRO.REC1 keywords. Therefore the extrapolation of a virtual RB is not possible. If available, the tool will recognise reduction Block file (.rb,.rX, etc) and use them to build associations.

An example of virtual reduction block is shown in Figure 16

Figure 16: snapshot of the **File Detail Panel,** showing a virtual reduction block reconstructed by *Gasgano* using the HIERARCH.ESO.PRO.REC1 keywords.

The meaning of various sections of the reduction block is discussed in section.2.2.

When more than one virtual reduction block is associated with a selected file, all components from all the reduction blocks are combined and listed in the registers. If one of these virtual reduction blocks is selected, the files associated with the selected reduction block will be displayed in bold in the **Raw Files and Products Files** registers.

In a virtual reduction block, names of files currently loaded by the application are displayed with their full path on the local file system. Names of files not currently loaded by the application are shown without path information. This is because the REC1 keywords store only file names, and contain no path information (as the path information is not fixed).

Upon selection of a file in the **Navigation Panel**, files from the various sections of the virtual reduction block will be listed in the different registers of the **File Association Panel**:

- Register **Raw Files** lists all raw files used to create a product. Clicking on any of these entries sets the selection in the **Navigation Panel** to this file. If the file is not loaded a "File not loaded" message is displayed in the **File Details Panel**.

- Register **Master Calib. File**: lists the master calibration products used during data reduction. In the example in Figure 16, these files are: `IS_SSAM_010608A_SK_s03t_216_Xe+Ar.tfits` and `IS_SSFM_010608A_SK_s03t_216.fits`, both stored in the directory `/raid51_isaac/isaac/raw/gasgano_swap/test` and both currently loaded into the application. (In the case if a virtual reduction block, we know that these files are currently loaded in the application as they are displayed with the full path information).

- Register **Reduction Block Files**: lists all reduction blocks associated with the selected file. Clicking on any entry listed in this register will display the reduction block in the **File Details Panel**.

- Register **Product Files**: lists all products associated with the selected file. Clicking on any file listed in the **Product Files** register selects this file in the **Navigation Panel**, if the file is currently loaded in the application.

The button **Save to File** in the **File Details Panel** allows the user to save the currently displayed virtual reduction block to disk.

It is possible to save and subsequently edit virtual reduction blocks from within the application:

- Save the virtual reduction block to a file.

- Load the file into *Gasgano* (or simply refresh the view, if it was saved in currently loaded directory).

- Edit it using an editing script from the **Script Board**. (E.g. a shell script that calls you favourite ASCII file editor).

7 *ESO – THE DATABASE VIEW*

TBD

8 *REFERENCE GUIDE*

8.1 Summary of Inputs

Table 1: Keyboard shortcuts

SHORTCUT	ACTION
CTRL+q	Quit Gasgano
CTRL+r	Open keyword list editor
CTRL+s	Toggle open/close the Script Board
CTRL+c (in the File Detail Panel)	Copy content of selection to
CTRL+v (in a text filed, such as	Paste the content of the Clipboard

Table 2: Mouse actions

BUTTON	CONTEXT	EFFECT
Left click	Navigation panel, on nodes, folder icons, folder lines.	Expand the corresponding item, showing its sub-components
Left click	Navigation panel , on files	Select the file
Right click	Navigation panel, on nodes, folder icons, folder lines.	Selection of all files included in the folder and its subfolders
Right click	Keyword lists editors (also in **Preferences** editor) onto **Add/Insert Row** button	Add/insert empty row above selected row
Ctrl-Left click	Navigation panel , on files	Multiple selection of non contiguous files
Shift-Left click	Navigation panel , on files	Multiple selection of contiguous files selection (first file must be already selected)
Ctrl-Right	Navigation panel, on nodes, folder icons, folder	Multiple selection of content of

click	lines.	non contiguous folders (nodes)
Shift-Right click	Navigation panel, on nodes, folder icons, folder lines.	Multiple selection of content of contiguous folders (nodes)
Middle mouse button	Navigation panel, on nodes, folder icons, folder lines.	Save the content of the folder line into an internal clipboard. The content of the clipboard can be then used via the Script Board. It emits a beep.

8.2 Keywords

Gasgano uses keywords for expressions, classification, display and report generation. There are not keyword dictionaries or defined keyword sets in *Gasgano* and entries of non-existent or bad keyword names will not be detected. For example a FITS file report requiring the `PIPPO` keyword to be printed will not produce an error. The column in the resulting report will be left blank.

ESO Hierarchical keywords can be expressed in both "full" (e.g.: `HIERARCH ESO TEL AIRM START`) and "dot" (e.g.: `TEL.AIRM.START`) notation. While *Gasgano* supports both forms of notation for data entry, typically the "dot" notation is shown for display purposes.

8.3 Meta Keywords

Gasgano defines internally a number of keywords, which are used within the application but are not read from a FITS file header. These keywords are called Meta Keywords and are designed to never conflict with keywords read from a FITS file. This is achieved by giving all Meta Keywords "illegal" names.

Meta Keywords are used to represent data which the user may want to use in Keywords expressions and reports, but which is not contained within the current set of keywords read from a FITS header.

The Meta Keywords currently in use in Gasgano are the following:

- `CLASSIFICATION`: a keyword representing the FITS file Classification, as defined by means of the classification rules. The keyword is "illegal" since it is not hierarchical and it is over 8 characters in length.
- `FILE_PATH`: a keyword representing the full path of the data file on disk. This keyword is "illegal" as it contains an underscore ('_') character.
- `PIPE_PRODUCT`: a keyword indicating if the file is a raw frame or pipeline product. A raw frame will have a `PIPE_PRODUCT` value of "F" while a pipeline product file will have a `PIPE_PRODUCT` value of "T". This keyword is "illegal" as it contains an underscore ('_') character.

8.4 Keyword Expressions

This section discusses the use of keyword expressions in *Gasgano*. Keyword expressions are used to classify FITS data files and filter the data file set.

Keyword expressions evaluate to a boolean value. They are a combination of Operands and Operators.

Operands have three different types: character strings, numbers and keywords.

- Keywords are represented as the names of the keyword and return either a Float or a String value.
- Strings are represented as characters enclosed in double quotes (e.g. “SKY”). A String can not return a Float value.
- Floats are numeric values, (12, 0.1, 45.67, etc.). Floats can not return a String value.

Supported Operators are listed in Table 3. (LHS = Left Hand Side; RHS = Right Hand Side)

Table 3: Keyword expression operators

OPERATOR	DESCRIPTION
AND	Boolean operator which evaluates to **true** if and only if all of its two or more sub-expressions evaluate to true.
OR	Boolean operator that evaluates to **true** if one of its two or more sub-expressions evaluates to true.
=	Evaluates to **true** if the LHS equals the RHS.
!=	Evaluates to **false** if the LHS equals the RHS.
>	Evaluates to true if the LHS is greater than the RHS.
>=	Evaluates to true if the LHS is greater or equal to the RHS.
<	Evaluates to true if the LHS is less than the RHS.
<=	Evaluates to true if the LHS is less than or equal to the RHS.
Like	Like performs a string comparison where the RHS must be a string operand. The RHS may contain '*' characters which are treated as wild cards when matching with the LHS. **Note: Only the "*" wild card is valid, there are no other wild card operators available with the Like operator.**
NotLike	As for Like but the opposite result is returned upon evaluation.

8.5 Keyword Expression Classes

A keyword expression is a tree structure with AND and OR operators as nodes and comparison operators (=, >, <, Like, etc) as leaves of the tree structure. Evaluation of the expression involves evaluation of each of its leaves and nodes in a depth first traversal until a result for the root node is determined.

The nodes of the tree are either AND or OR operators containing two or more branches. The leaves of the nodes must be comparison operators (=, !=, <, etc.). Each comparison operator contains a RHS (Right Hand Side value) and LHS (Left Hand Side value) and an operator.

Comparison operators may be evaluated as Float objects or String objects depending on the values of operands. A comparison will be evaluated using Float objects only if all operands return a valid Float value. Failing this, the comparison is tried using String objects. If comparison using Floats or Strings is not possible, the expression will fail to evaluate.

Figure 17 shows a couple of simple expressions and their string representations.

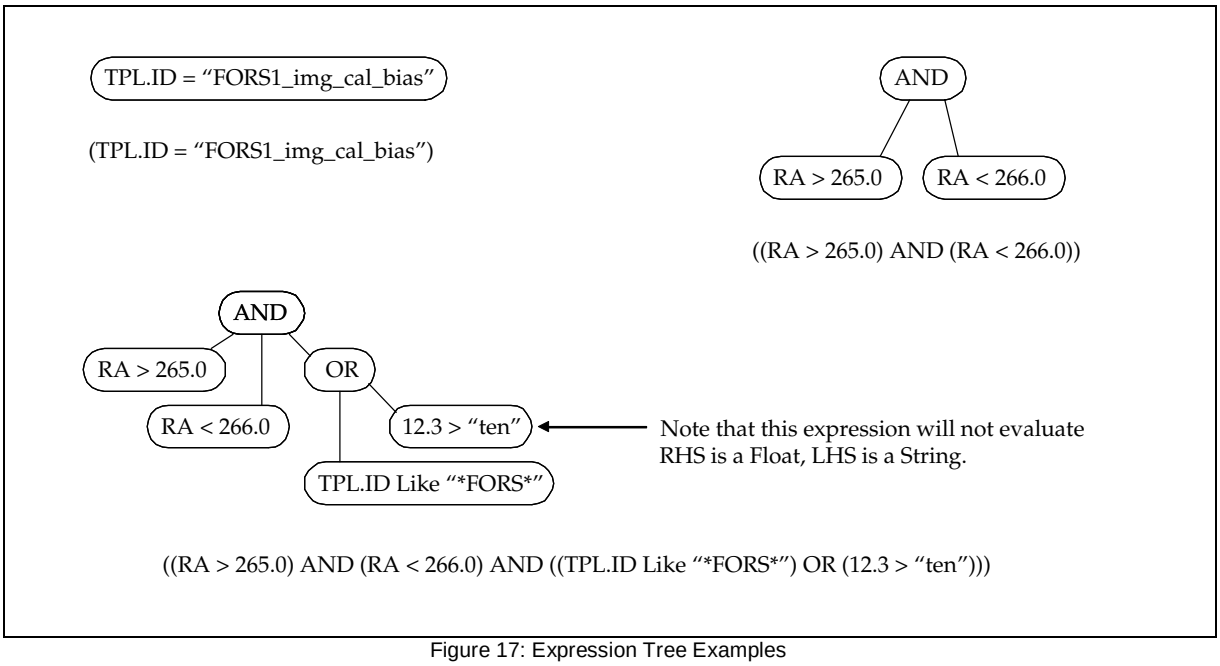

Figure 17: Expression Tree Examples

9 *APPENDIX A - .gasganorc SETTINGS*

9.1 Printout of .gasganorc (19.06.2001)

```
#Gasgano Properties Editor
CLASSRULE_FILE=rules.rul
COMPONENT_FONT=Dialog:14
CUBE_TO_VIEWER=
DATA_FILES=
DBOBS_DISPLAY_COLUMNS=CLASSIFICATION:RA:DEC:TPL.EXPNO:TPL.NEXP
DBOBS_DISPLAY_COLUMNS_SIZE=0:0:0:0:0
DBOBS_INSTRUMENT=All
DBOBS_INSTRUMENTS=ISAAC;FORS1;FORS2;UVES;VIMOS;CONICA;EFOSC;CES;SUSI;
SOFI;EMMI;WFI;VINCI;MIDI
DBOBS_PERIOD=66
DBOBS_REQUIRED=false
DBOBS_SERVER_ID=
DBOBS_SERVER_URL=
DBOBS_USERID=
DB_SERVER_URL=
DB_USER_NAME=
DISPLAY_COLUMNS=CLASSIFICATION:TPL.ID:ORIGFILE:TPL.EXPNO:TPL.NEXP
DISPLAY_COLUMNS_SIZE=0:0:0:0:0
FILTER_FILE=filter.rul
FITS_TO_VIEWER=
FRAME_BOUNDS=8,100,1270,800
INSTRUMENT_GROUPING=ISAAC=;FORS1=;FORS2=;UVES=;VIMOS=;CONICA=;EFOSC=;
CES=;SUSI=;SOFI=;EMMI=;WFI=;VINCI=;MIDI=
KEYWORDLIST_DIR=gas-keywords
MENU_CUSTOM1=
MENU_CUSTOM1_LBL=Custom  Menu1
MENU_CUSTOM2=
MENU_CUSTOM2_LBL=Custom  Menu2
MENU_CUSTOM3=
MENU_CUSTOM3_LBL=Custom  Menu3
MENU_CUSTOM4=
MENU_CUSTOM4_LBL=Custom  Menu4
MENU_RETRIEVE=dpclient  -renameschema ARCFILE -outpath
MJD_SORTING=false
OBSTATUS_MAPPING=P:Partially defined D:Defined Q:queued for
scheduling I:Initiated
 S:Started p:Paused during execution A:Aborted X:Executed R:Reduced
C:Completed L:
released K:Cancelled
OB_STATUS_FROM_DB_REQUIRED=false
PRINTER_NAME=
PRINT_FONT=SansSerif:16
PRINT_ORIENTATION=P
RADEC_CONVERSION=true
SCRIPTS_DIR=scripts
SHORTEN_FILES_PATH=true
SHORT_FILENAME=true
TEXTFILE_EXTENSIONS=ascii;paf
TFITS_TO_VIEWER=
UPDATE_TIMER=0
VERBOSE_LEVEL=0
```

The following table contains the correspondence between entries in the **Preference Editor Panel** and those in the .gasganorc file.

Field in preference editor	Entry in resource file .gasganorc	User's guide
Register Data		
Data_Directories_and_files with button [Add File] and button [Remove Selected Files]	DATA_FILES	The list of directories or files automatically loaded at application startup
"Keyword List directory:"	KEYWORDLIST_DIR	The default directory where the application looks for keyword list files
"Scripts directory:"	SCRIPTS_DIR	The directory where the application looks for executable external commands
"Filter_Rule file:"	FILTER_FILE	The file containing the rules for Navigation Panel filtering (Filesystem view)
"Classification Rule file:"	CLASSRULE_FILE	The file containing the classification rules
"Textfile_Extensions:"	TEXTFILE_EXTENSIONS	Recognised file extensions for ASCII files
"Shorten data files path for commands:"	SHORTEN_FILES_PATH	True/false: specifies whether command piped to the shell should be shortened
"Automatic Filessystem Refresh:"	UPDATE_TIMER	1,2,5,30 frequency for automatic filesystem update
"Verbose_Level:"	VERBOSE_LEVEL	Category of messages to be verbosed
Register Databases		
Database:Read_observations_from_Database:	DBOBS_REQUIRED	True/false: (ESO) : whether the database view is active
Database:server URL	DBOBS_SERVER_URL	The fully qualified hostname and port for the database server containing the observations database
Database:server ID	DBOBS_SERVER_ID	An identifier for the server containing the observations database
Database:username	DBOBS_USERID	The userid to log into the abovementioned database
OB_status:Display OB status (from database)	OB_STATUS_FROM_DB_REQUIRED	True/false (ESO): whether the Observation Block status is displayed for each Observation
OB_status:server_URL	DB_SERVER_URL	The fully qualified hostname and port for the database server containing the obrep2 database
OB_status:username	DB_USER_NAME	The userid to log into the abovementioned database
OB-statusMapping	OBSTATUS_MAPPING=	A simple mapping between one-character return codes and human readable descriptions for the OB status
Register Display&Printing		
Filesystem_Displayed_Keywords	DISPLAY_COLUMNS	List of semicolon-separated keywords displayed in the navigation panel for filesystems
Printer_Name	PRINTER_NAME	Name of printer
Printer_orientation	PRINT_ORIENTATION	L/P: Landscape or Portrait
Printer_font	PRINT_FONT	The font used for printing

Appendix A - .gasganorc settings

Field in preference editor	Entry in resource file .gasganorc	User's guide
Font Test	----	
Tfits_Viewer	TFITS_TO_VIEWER	Application to be executed to visualise tfits files
Fits_Viewer	FITS_TO_VIEWER	Application to be executed to visualise fits files
Cube_viewer	CUBE_TO_VIEWER	Application to be executed to visualise a set if fits files
truncate_filenames	SHORT_FILENAME	True/false: whether the filename in the navigation panel should be specified with its complete path. For the filesystem only
convert_RA/REC	RADEC_CONVERSION	Whether RA and DEC keywords are visualized in sexagesimal notation from the original decimal
Sort by MJD	MJD_SORTING	Whether fits and tfits files in the navigation panel are sorted acording to they MJD or according by name
Database_Displayed_keywords	DBOBS_DISPLAY_COLUMNS	List of semicolon-separated keywords displayed in the navigation panel for the database
Register Menus&Instruments		
"instrument specific intermediate grouping (filesystem only)"	-----	
IntermediateGroupTable	INSTRUMENT_GROUPING	List of semicolon separated pairs of INSTRUMENT=KEYWORD to be used for further instrument grouping
"database pull-down menus associations"	-----	
database_pull_down_menus.raw1.col2	MENU_CUSTOM1_LBL	Label for custom menu1 – database only
database_pull_down_menus.raw1.col3	MENU_CUSTOM1	Application bound to menu item 1
database_pull_down_menus.raw2.col2	MENU_CUSTOM2_LBL	Label for custom menu2 – database only
database_pull_down_menus.raw2.col3	MENU_CUSTOM2=	Application bound to menu item 2
database_pull_down_menus.raw3.col2	MENU_CUSTOM3_LBL	Label for custom menu3 – database only
database_pull_down_menus.raw3.col3	MENU_CUSTOM3=	Application bound to menu item 3
database_pull_down_menus.raw4.col2	MENU_CUSTOM4_LBL	Label for custom menu4 – database only
database_pull_down_menus.raw4.col3	MENU_CUSTOM4	Application bound to menu item 4
database_pull_down_menus.raw5.col3	MENU_RETRIEVE	Application to retrieve a file from the databaseaime
The following information are not available via the PreferenceEditor Panel.		
-----	FRAME_BOUNDS	Format: X,Y,Width, Height The size and location of the main frame when the preferences are accepted
-----	DISPLAY_COLUMNS_SIZE	The size in points of the columns of the Navigation Panel /Filesystem view – Reserved for future use
-----	DBOBS_INSTRUMENT	The instrument selected for the database query in the database view
-----	DBOBS_INSTRUMENTS	The list of all possible instruments which can be used in the database view

Appendix A - .gasganorc settings

Field in preference editor	Entry in resource file .gasganorc	User's guide
-----	DBOBS_DISPLAY_COLUMNS_SIZE	The size in points of the columns of the Navigation Panel /Database view – Reserved for future use
-----	DBOBS_PERIOD	The observing period selected for the query in the database view
-----	COMPONENT_FONT	Reserved for future use

10 APPENDIX B – VERBOSE LEVELS

The verbose level can be set in the Data **register of** Preferences… Menu (see section 4.1.1. The verbose facility is to be used for debugging purposes only, upon request from ESO Support personnel and is in no way meant to be complete or understandable for the users. The default value for this option should be set to **No Verbose** during normal use of the tool.

The possible values and their description are:

DataModelChange: changes like addition or removal of OBComponents andVRB to the application memory are displayed

DataModelEvent: notification to the Navigation Panel tree of addition/removal of components

ExternalCommands: not used

BaseName: verboses information on how the basename for an OBComponent isdetermined

Timing: activates collection of statistics for header read,classification and overall file read times.

Events: redundant, to be deleted in future releases

ExpressionAndRules: verbose info concerning the classification rule editor and the visualization of Expressions as strings

Database: (ESO) all queries and database initialization procedures

All: the union of all the verbose levels